



**SURVIVABILITY - SUSTAINABILITY - MOBILITY  
SCIENCE AND TECHNOLOGY  
SOLDIER SYSTEM INTEGRATION**

**AD-A284 375**



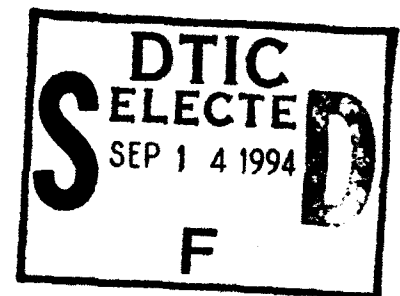
**TECHNICAL REPORT  
NATICK/TR-94/015**

AD \_\_\_\_\_

(1)

**PARACHUTE INFLATION:  
A PROBLEM IN AEROELASTICITY**

By  
**Keith R. Stein  
Richard J. Benney**



**94-29743**

**August 1994**



*86 PJ*

**FINAL REPORT  
January 1993 - February 1994**

**Approved for Public Release; Distribution Unlimited**

**94 9 13 024**

**UNITED STATES ARMY NATICK  
RESEARCH, DEVELOPMENT AND ENGINEERING CENTER  
NATICK, MASSACHUSETTS 01760-5000**

**MOBILITY DIRECTORATE**

**DTIC QUALITY INSPECTED 3**

### DISCLAIMERS

The findings contained in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of trade names in this report does not constitute an official endorsement or approval of the use of such items.

### DESTRUCTION NOTICE

#### For Classified Documents:

Follow the procedures in DoD 5200.22-M, Industrial Security Manual, Section II-19 or DoD 5200.1-R, Information Security Program Regulation, Chapter IX.

#### For Unclassified/Limited Distribution Documents:

Destroy by any method that prevents disclosure of contents or reconstruction of the document.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE AUGUST 1994	3. REPORT TYPE AND DATES COVERED FINAL Jan 1993 - Feb 1994	
4. TITLE AND SUBTITLE PARACHUTE INFLATION: A PROBLEM IN AEROELASTICITY			5. FUNDING NUMBERS WU 1L162786D283AJH00 1L161102AH5210B00 1L162786D283APA00	
6. AUTHOR(S) Keith R. Stein and Richard J. Benney				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Natick RD&E Center Kansas St. ATTN: SATNC-VU Natick, MA 01760			8. PERFORMING ORGANIZATION REPORT NUMBER NATICK/TR-94/015	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <p>In parachute research, canopy inflation is the least understood and most complex process to model. Unfortunately, it is during the opening process that the canopy experiences the largest deformations and loadings. The complexity of modeling the opening process stems from the coupling between the structural dynamics of the canopy, lines plus payload and the aerodynamics of the surrounding fluid medium. The addition of a computational capability to model the coupled opening behavior would greatly assist in understanding the canopy inflation process. Ongoing research at the U.S. Army Natick Research, Development and Engineering Center (Natick) focuses on this coupled problem. The solution to this problem will assist in the development of future U.S. Army airdrop systems, which include the capability of deploying at low altitudes and high speeds. This report describes research at Natick that currently involves coupling a computational fluid dynamics (CFD) code to a mass spring damper (MSD) parachute structural code. The model is described and results are presented.</p> <p style="text-align: right;">DTIC QUALITY INSPECTED 3</p>				
14. SUBJECT TERMS PARACHUTES      COUPLED MODELS      AEROELASTICITY CANOPIES      COMPUTATIONAL FLUID DYNAMICS      PARACHUTE INFLATION AIRDROP      MASS SPRING DAMPING SYSTEM			15. NUMBER OF PAGES 95	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT SAR	

# CONTENTS

	Page
FIGURES .....	v
TABLES .....	ix
PREFACE .....	xi
LIST OF SYMBOLS .....	xiii
INTRODUCTION .....	1
PROGRESSION OF NUMERICAL MODEL .....	2
Computational Fluid Dynamics (CFD) Model .....	2
SALE computer program .....	2
Decelerator surface defined by interior grid points .	3
Implementation of C-GRID .....	7
Mass Spring Damper (MSD) Model .....	11
Introduction to MSD model .....	11
MSD equations of motion .....	14
CALA logic and equations .....	16
MSD numerical approximations .....	19
Reformulation for SLATEC software .....	24
Initial conditions .....	25
Boundary conditions .....	26
Infinite mass opening .....	27
MSD Fortran program description .....	28
Coupling .....	31
Introduction .....	31
CFD Decelerator surface defined by interior grid points	31
Implementation of C-Grid .....	31
MATLAB & AVS: Preprocessing & Postprocessing .....	33
CFD preprocessing and postprocessing .....	33
MSD postprocessing .....	34
MSD MATLAB program description .....	34
AVS programs for MSD postprocessing .....	35
RESULTS .....	36
Case 1 .....	38
Case 2 .....	39
Case 3 .....	40
Case 4 .....	41
Case 5 .....	42
Case 6 .....	43

## CONTENTS (Cont'd)

DISCUSSION .....	44
Computational Fluid Dynamics Model .....	44
Mass Spring Damper Model .....	44
Coupling .....	45
CONCLUSIONS .....	45
REFERENCES .....	46
APPENDICES .....	49
A- Case 1 .....	51
B- Case 2 .....	57
C- Case 3 .....	61
D- Case 4 .....	65
E- Case 5 .....	69
F- Case 6 .....	73

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification .....	
By .....	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

## FIGURES

	Page
Figure 1. Computational Grid Structure (Interior Decelerator Surface) .....	4
Figure 2. Interior "Rezone" Region of Computational Grid ...	5
Figure 3. Boundary Conditions (Interior Decelerator Surface).....	6
Figure 4. Surface Pressure Extrapolation (Interior Decelerator Surface).....	7
Figure 5. C-GRID Computational and Physical Domains .....	8
Figure 6. Generation of C-Grid .....	10
Figure 7. Mass Spring Damper Model .....	12
Figure 8. Free Body Diagram .....	12
Figure 9. Definition of Angles .....	20
Figure 10. Layout for Two Half Gores .....	22
Figure 11. Meridional Spring Constants .....	23
Figure 12. Flow Chart Outline for MSD Code Subroutine .....	29
Figure A1. Canopy Shape Versus Time in Seconds ( $0.0 < t < 1.0$ ) ...	52
Figure A2. Canopy Shape Versus Time in Seconds ( $1.0 < t < 2.0$ ) ...	52
Figure A3. Canopy Shape Versus Time in Seconds ( $2.0 < t < 3.0$ ) ...	53
Figure A4. Payload Force, Velocity & Position Versus Time ....	53
Figure A5. Payload Force Versus Time (Numerical & Experimental)	54
Figure A6. Payload Force Versus Time (Two Different Cn's) ....	54
Figure A7. Payload Velocity Versus Time (Two Different Cn's)	55
Figure A8. Payload Position Versus Time (Two Different Cn's)	55
Figure B1. Canopy Shape Versus Time in Seconds ( $0.0 < t < 1.0$ ) ...	58
Figure B2. Canopy Shape Versus Time in Seconds ( $1.0 < t < 1.3$ ) ...	58
Figure B3. Canopy Shape Versus Time in Seconds ( $1.3 < t < 2.3$ ) ...	59

## FIGURES (Cont'd)

Figure B4.	Payload Force, Velocity & Position Versus Time ....	59
Figure C1.	Canopy Shape Versus Time in Seconds ( $3.0 < t < 3.1$ ) ...	62
Figure C2.	Canopy Shape Versus Time in Seconds ( $3.1 < t < 3.2$ ) ...	62
Figure C3.	Canopy Shape Versus Time in Seconds ( $3.2 < t < 3.3$ ) ...	63
Figure C4.	Payload Force & Velocity Versus Time .....	63
Figure C5.	Projected Diameter Over Constructed Diameter Versus Time .....	64
Figure D1.	Canopy Shape Versus Time in Seconds ( $0.0 < t < 5.0$ ) ...	66
Figure D2.	Payload Force, Velocity & Position Versus Time ....	66
Figure D3.	Images of 3-D Canopy Shapes .....	67
Figure E1.	Canopy Shape Versus Time in Seconds ( $0.0 < t < 10.0$ ) ..	70
Figure E2.	Payload Force, Velocity & Position Versus Time ....	70
Figure E3.	Images of 3-D Canopy Shapes .....	71
Figure F1.	Canopy Shape Versus Time in Seconds ( $1.6 < t < 2.0$ ) ...	74
Figure F2.	Canopy Shape Versus Time in Seconds ( $2.0 < t < 2.5$ ) ...	74
Figure F3.	Canopy Shape Versus Time in Seconds ( $2.5 < t < 3.5$ ) ...	75
Figure F4.	Canopy Shape Versus Time in Seconds ( $2.5 < t < 5.0$ ) ...	75
Figure F5.	Payload Force, Velocity & Position Versus Time ....	76
Figure F6.	Payload Force Versus Time (Numerical & Experimental)	76
Figure F7.	Payload Velocity vs Time (Numerical & Experimental)	77
Figure F8.	Canopy Area vs Time (Numerical & Experimental) ....	77
Figure F9.	CFD Solution ( $t=1.58$ ) .....	78
Figure F10.	CFD Solution ( $t=1.84$ ) .....	78
Figure F11.	CFD Solution ( $t=2.11$ ) .....	79
Figure F12.	CFD Solution ( $t=2.37$ ) .....	79

## FIGURES (Cont'd)

Figure F13. CFD Solution ( $t=2.63$ ) .....	80
Figure F14. CFD Solution ( $t=2.89$ ) .....	80
Figure F15. CFD Solution ( $t=3.16$ ) .....	81
Figure F16. CFD Solution ( $t=3.42$ ) .....	81
Figure F17. CFD Grid ( $t=1.58$ & $1.84$ ) .....	82
Figure F18. CFD Grid ( $t=2.11$ & $2.37$ ) .....	82
Figure F19. CFD Grid ( $t=2.63$ & $2.89$ ) .....	83
Figure F20. CFD Grid ( $t=3.16$ & $3.42$ ) .....	83



## TABLES

	Page
Table 1. Summary of Input Parameters .....	36

## PREFACE

This report describes the efforts undertaken as part of the projects "Fluid Structure Coupling During Parachute Opening" 1L161102AH5210B00, "Computational Prediction of the Optimum Shape for Aerodynamic Decelerators" 1L162786D283AJH00, and "Nonlinear Structural Dynamic Behavior of Parachutes", 1L162786D283AFA00. This progress report incorporates effort that was undertaken during the period January 1993 through February 1994. This work was performed by the Applied Research Division (ARD) of the Mobility Directorate (MobD). The authors would like to express their appreciation to Dr. Earl Steeves and Dr. Calvin Lee of ARD for their assistance in this effort. The authors would also like to express thanks to Mr. Ed Giebutowski and Mr. John Watkins of the Parachute Engineering Branch, Airdrop Systems Division, Mobility Directorate for their contributions of experimental data and video.

## LIST OF SYMBOLS

A	cross-sectional area of lines
CALAX	X-direction force contribution from CALA logic
CALAY	Y-direction force contribution from CALA logic
$Cm_1$	meridional damping constant
$Cn_1$	normal damping constant
$DX_1$	X-direction force contribution from normal line drag
$DY_1$	Y-direction force contribution from normal line drag
$E_1$	Young's modulus of lines
$E_2$	Young's modulus of canopy fabric
F1	normal force contribution
F2, F3	tangential force contribution
F4	gravitational force contribution
I	internal energy
N	total number of gores
P, Q	C-Grid forcing functions
g	gravitational constant
h	canopy fabric thickness
$km_1$	meridional spring constant
$lo_1$	constructed distance between nodes $i$ and $i+1$
$\Delta l_1$	positive amount of stretch between nodes $i$ and $i+1$
$m_1$	mass associated with node $i$
$m_{pay}$	payload mass
n	total number of nodes on canopy
$n_l$	total number of nodes on line
$n_x$	width of CFD grid in cells
$n_y$	height of CFD grid in cells
p	pressure
t	time
u	radial CFD component
v	axial CFD component
$v_{sp}$	variable used in infinite mass openings
$v_t$	time of inflection for infinite mass openings
$v_\infty$	infinite mass terminal velocity
x	radial fluid coordinate
$xo_1$	constructed meridional length to node $i$
$x(i)$	current x location of node $i$
y	axial fluid coordinate
$y(i)$	current y location of node $i$
$\alpha_1$	angle defining normal direction of node $i$
$\beta_1$	angle defining relative angle between nodes $i$ and $i+1$
$\eta$	computational domain coordinate
$\mu$	fluid viscosity
$\xi$	computational domain coordinate
$\pi$	3.14159...
$\rho$	fluid density
$\phi$	$2\pi/N$
$\psi$	gore bulge angle

# PARACHUTE INFLATION: A PROBLEM IN AEROELASTICITY

## INTRODUCTION

The time-variant aeroelastic characteristics associated with the opening of a parachute are extremely complex to model. The complexity of the problem arises because the flow field is dependent on the canopy shape, which is itself dependent on the flow field. A correct model must include the coupled behavior of the structural dynamics of the parachute system with the aerodynamics of the surrounding flow field. A coupled model will provide not only information about the opening characteristics of a parachute, but also characteristics of the parachute in its terminal velocity state including the parachute's shape, drag, velocity, pressure distribution, and flow-field characteristics.

Previously, either the aerodynamic or the structural dynamic behavior of the parachute opening problem was studied independently (decoupled). A variety of decoupled models developed and investigated at U.S. Army Natick Research, Development & Engineering Center (Natick) have contributed directly to the coupled model presented in this paper. These studies include steady and unsteady computational fluid dynamic (CFD) solutions about rigid decelerators [1,2,3]. Unsteady CFD solutions about decelerators with a specified opening behavior have also been investigated [4].

The logic required to couple a CFD code to a structural dynamic code was established in stages of increasing complexity. All models described in this report are axisymmetric models. The early stages of the model were presented in [5,6]. The present model involves coupling a CFD code to a mass spring damper (MSD) structural dynamic code representing a flat, circular solid-cloth parachute such as a C-9. This model is used in an attempt to predict the behavior of a variety of parachutes with varying initial conditions. A half-scale C-9 canopy dropped from rest was modeled and the computational results are compared with experimental results. Other simulations are compared with experimental results, including a reefed T-10 flat extended skirt parachute and a reefed G-12 cargo parachute. This report describes the current coupled model and presents computational results from the model during different stages of the codes progression. Future directions and potential enhancements are discussed.

## PROGRESSION OF NUMERICAL MODEL

### Computational Fluid Dynamic (CFD) Model

#### SALE computer program

The Simplified Arbitrary Lagrangian-Eulerian (SALE) code, written at Los Alamos National Laboratories, has been adapted to solve the Navier-Stokes equations about aerodynamic decelerator shapes [7]. SALE uses a finite difference algorithm to solve the time-dependent two-dimensional Navier-Stokes equations in Cartesian or axisymmetric coordinates. Axisymmetric coordinates are used for parachute applications. The time-dependent, axisymmetric, incompressible Navier-Stokes equations are shown in equations (1)-(3).

Conservation of Mass-

$$\frac{\partial \rho}{\partial t} + \frac{1}{x} \frac{\partial x \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = 0 \quad (1)$$

Conservation of Momentum-

$$\frac{\partial \rho u}{\partial t} + \frac{1}{x} \frac{\partial x \rho u^2}{\partial x} + \frac{\partial \rho uv}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{x} \frac{\partial x \pi_{xx}}{\partial x} + \frac{\partial \pi_{xy}}{\partial y} - \frac{\pi_{\theta}}{x} \quad (2)$$

$$\frac{\partial \rho v}{\partial t} + \frac{1}{x} \frac{\partial x \rho uv}{\partial x} + \frac{\partial \rho v^2}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{x} \frac{\partial x \pi_{xy}}{\partial x} + \frac{\partial \pi_{yy}}{\partial y}$$

Conservation of Internal Energy-

$$\frac{\partial \rho I}{\partial t} + \frac{1}{x} \frac{\partial x \rho I u}{\partial x} + \frac{\partial \rho I v}{\partial y} = -pD + \pi_{xx} \frac{\partial u}{\partial x} + \pi_{xy} \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + \pi_{yy} \frac{\partial v}{\partial y} + \frac{u \pi_{\theta}}{x} \quad (3)$$

where,

$$D = \frac{1}{x} \frac{\partial x u}{\partial x} + \frac{\partial v}{\partial y}$$

and,

$$\pi_{xx} = 2\mu \frac{\partial u}{\partial x}$$

$$\pi_{yy} = 2\mu \frac{\partial v}{\partial y}$$

$$\pi_{xy} = \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)$$

$$\pi_{\theta} = 2\mu \frac{u}{x}$$

SALE defines velocities at cell vertices in the computational grid, whereas pressures are defined at cell centers. SALE uses the Arbitrary Lagrangian-Eulerian (ALE) algorithm, which allows use of nonuniform computational grids which can deform with time. The computational domain is discretized with a single block grid consisting of quadrilateral cells. The rezoning capabilities of SALE are valuable for solving flows about decelerators in motion or for inflation problems. The solution algorithm of SALE is broken into three phases. The first phase of the solution is a purely explicit computation, in which the velocity field is updated by accounting for the effects of all forces. Pressure, viscous and other forces are taken into account during this first phase. The second phase of the solution is implicit and advances the pressures and velocities in time with a Newton-Raphson iteration. This implicit phase allows for larger timesteps, and thus greater efficiency is possible for low speed and incompressible flows. The updated velocity field, from the first phase of the solution, is used as the initial guess in the Newton-Raphson iteration. The iterative process continues until an error tolerance for the pressure field is obtained. The second phase is bypassed for purely explicit calculations of SALE (all problems presented utilize explicit-implicit computations). The final phase of the SALE solution algorithm is an advective flux calculation. This calculation accounts for the change in individual cell properties due to advection across cell faces. For a purely Lagrangian computation (in which cell vertices move precisely with the fluid motion) there is no advection across cell faces and the advective flux calculation is skipped. However, for purely Eulerian computations (in which the grid vertices are fixed) or for arbitrary rezoning schemes, advection of fluid across cell faces must be accounted for. It is this final phase of the solution algorithm that makes SALE suited for parachute problems. The arbitrary rezoning capability allows for special mesh update strategies to be developed that can deform the computational mesh about an opening canopy in time. SALE has been modified to solve various parachute flow problems such as steady and unsteady flow about rigid decelerators [1,2,3], unsteady flow behavior about parachute shapes with specified opening behaviors [4], and coupled fluid-structure interaction problems for opening parachutes [5,6].

#### Decelerator surface defined by interior grid points

SALE computations for rigid decelerators, for decelerators with specified openings, and initial coupled fluid-body interaction computations utilized a mesh in which the decelerator surface was defined by a single row of adjacent interior grid points (see figure 1). In order to perform computations for deforming decelerators, a mesh update strategy is needed that deforms the computational grid in time so that the interior "decelerator" grid points are repositioned on the decelerator each timestep and surrounding grid points are updated about the new shape. The

computational grid consists of an interior "rezone" region which includes the decelerator grid points and an exterior region which is rectilinear and has a less dense grid structure than the "rezone" region (see figure 2).

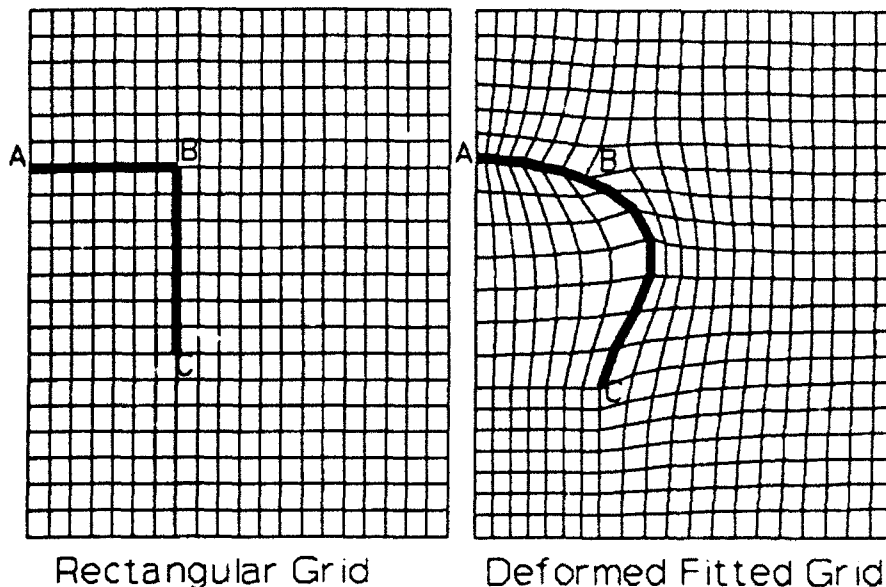


Figure 1. Computational Grid Structure (Interior Decelerator Surface)

An initial grid is created by first generating the interior "rezone" region and then extending the exterior region algebraically from the outer boundary of the "rezone" region. The interior "rezone" region is generated by deforming an initially uniform, rectangular grid so that appropriate cell vertices fit the desired decelerator shape. Positions of interior grid points that are not on the decelerator surface are determined iteratively by solving the Laplace equation with some algebraic manipulation to minimize distortion of the mesh. This gridding approach is demonstrated in figure 1 where C is the grid vertex representing the skirt of the canopy and A represents the canopy apex.

To fit the time-variant canopy shape, the interior "rezone" region of the grid is updated each time step with the same procedure used to create the initial grid. The exterior region is updated to extend from the new distribution of grid points along the outer boundary of the "rezone" region. Once all grid points have been repositioned, the relative motion of the grid and the corresponding fluid is determined at each grid vertex. During each time step, boundary conditions are imposed for the four outer boundaries of the computational grid and for the decelerator

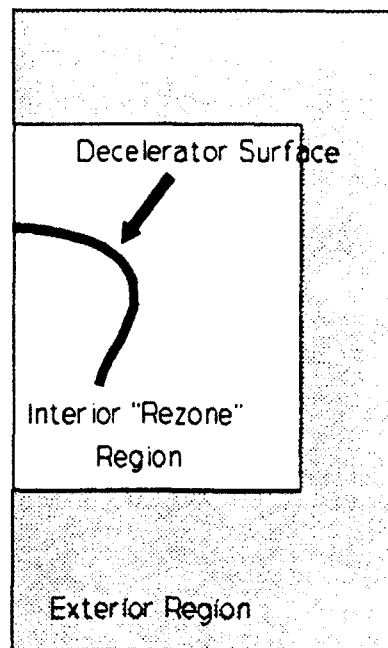


Figure 2. Interior "Rezone" Region of Computational Grid

surface. The boundary condition imposed on the decelerator surface is governed by the canopy motion. Nodal velocities are set to the corresponding canopy velocities to represent a noslip surface. The lower inflow boundary is considered the undisturbed, far field and both components of velocity are set to zero. The left boundary (or symmetry axis) and right boundary are given free-slip conditions which requires that there is no normal velocity component. The upper outflow boundary requires that the normal velocities are scaled in order to satisfy conservation of mass within the computational domain. The boundary conditions are depicted graphically in figure 3.

In coupled fluid-structure interaction computations, the structural dynamic code modeling the parachute requires nodal pressure differences as input at all vertices on the canopy surface. Since SALE computes pressure values at cell centers, vertex pressures are defined as the average of the surrounding four cell pressures. Values for the surface pressures are then extrapolated from the two neighboring vertices. This process is done for both the inner and outer surface of the canopy. This is shown in figure 4 where subscripts P1 and P2 are vertex pressures and P3 is the extrapolated nodal pressure on the upper surface of the canopy.

Point B in figure 1 is on the canopy surface at a corner in the CFD grid and often produces an adjacent cell which is quite distorted. The result is a pressure distribution with a slight



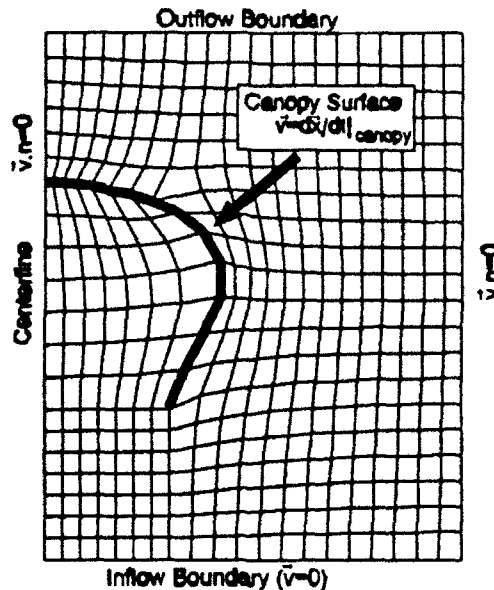


Figure 3. Boundary Conditions (Interior Decelerator Surface)

discontinuity at point B. For this reason, the differential pressure value sent to the MSD model at point B is evaluated by interpolation from a curve fitting the differential pressures at the surrounding vertices.

The mesh updating strategy implemented in SALE that is described above was initially coupled with a spherical membrane representation for the parachute canopy [15]. Later, it was coupled with an axisymmetric mass-spring-damper parachute model [5]. Computations with these coupled fluid-structure interaction models will be described later.

The initial mesh updating strategy is able to produce results for parachute inflation problems; however, the gridding approach has several limitations. First, discontinuities result from two particular points in the computational grid, the canopy surface node on the corner of the undeformed grid and the point on the skirt of the canopy (see points B and C in figure 1). A second limitation in the described gridding approach is a result of using the same set of nodes to define the canopy shape throughout the entire canopy inflation process. An adequate grid about the initial canopy shape will result in a final grid that is quite skewed. Likewise, in order to have a good grid about the final canopy shape a skewed initial grid must be used. Very little can be done to obtain grids that are nearly orthogonal throughout the complete computation. The gridding strategy has another limitation. The combination of having a rectangular grid and an

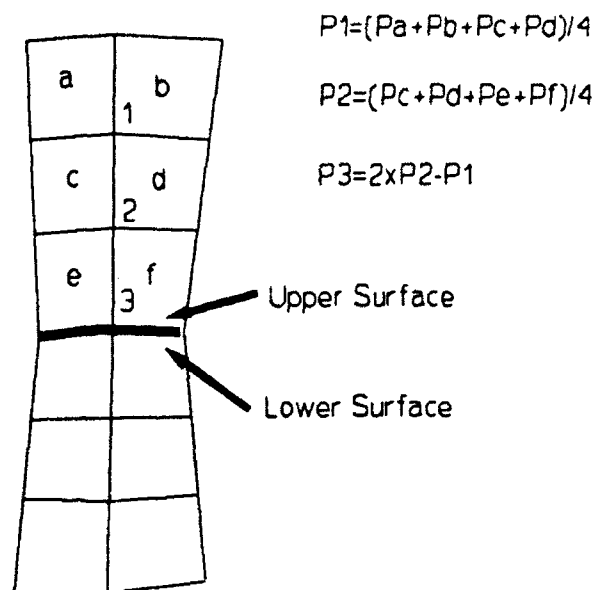


Figure 4. Surface Pressure Extrapolation (Interior Decelerator Surface)

interior surface makes clustering of grid points and coordinate control very limited. Grid spacing must be of the same order of magnitude throughout the grid. Thus, boundary layer resolution is impossible with the described gridding approach. Finally, SALE uses a single-block structured grid which imposes limits on the quality of the computational grid that can be generated for parachute problems.

#### Implementation of C-Grid

In order to address some of the gridding limitations stated above, an alternate gridding scheme has been implemented. SALE requires that the new gridding scheme remain a single-block grid with quadrilateral cells. An elliptic "C-Grid" mesh update strategy was employed. Much effort has been devoted to elliptic grid generation and some of the developed techniques and strategies could be transferred to the axisymmetric parachute model using C-Grids [8,9,10]. Elliptic grid techniques often result in grids that are nearly orthogonal, smooth, and have coordinate control options such as clustering towards specified boundaries.

A C-Grid takes a single block grid in the computational domain and maps one of its outer boundaries around the surface of a body in the physical domain. In the case of the current application, the boundary wraps around the parachute cross-section (see figure 5). The two boundaries that border the surface boundary in the

computational domain map onto the symmetry axis above and below the parachute surface boundary in the physical domain. The final boundary in the computational domain corresponds to the outer boundary in the physical domain.

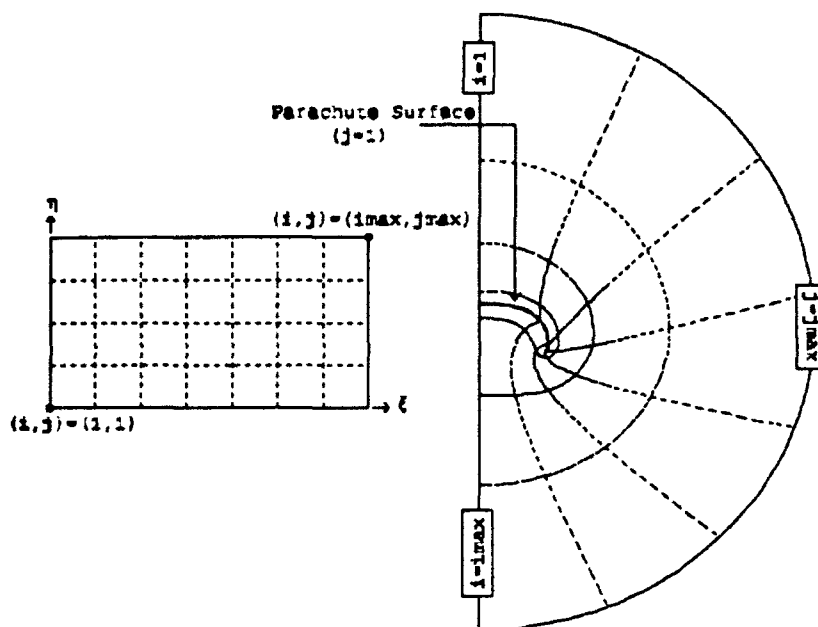


Figure 5. C-Grid Computational and Physical Domains

Elliptic grids require that the positions of all boundary points are known. Interior points in the grid are then determined iteratively by solving a system of elliptic equations in the form of Poisson's equation over the domain. A general form of Poisson's equation is shown in equations (4) and (5) below:

$$\xi_{xx} + \xi_{yy} = P(\xi, \eta) \quad (4)$$

$$\eta_{xx} + \eta_{yy} = Q(\xi, \eta) \quad (5)$$

where  $P$  and  $Q$  are "forcing function" that result in desired grid coordinate control. The transformation of equations (4) and (5) by interchanging the dependent variables ( $x$  and  $y$ ) with the independent variables ( $\xi$  and  $\eta$ ) leads to equations (6) and (7).

$$\alpha \xi_{\xi\xi} - 2\beta \xi_{\xi\eta} + \gamma \xi_{\eta\eta} = -J^2 (P\xi_{\xi} + Q\xi_{\eta}) \quad (6)$$

$$\alpha Y_{\xi\xi} - 2\beta Y_{\xi\eta} + \gamma Y_{\eta\eta} = -J^2 (PY_{\xi} + QY_{\eta}) \quad (7)$$

where,

$$\begin{aligned} \alpha &= x_{\eta}^2 + y_{\eta}^2 \\ \beta &= x_{\xi}x_{\eta} + y_{\xi}y_{\eta} \\ \gamma &= x_{\xi}^2 + y_{\xi}^2 \\ J &= x_{\xi}y_{\eta} - x_{\eta}y_{\xi} \end{aligned}$$

An initial algebraic grid is defined by specifying the decelerator surface at row  $\eta=1$  and the outer boundary at  $\eta=jmax$  in the computational domain. The remaining points are equally spaced along  $\xi=\xi_1$  for  $i=1$  to  $imax$ . Boundaries  $\xi=1$  and  $\xi=imax$  define the axis of symmetry in the physical domain.

Interior points and axis of symmetry points are determined by solving Poisson's equation using a Gauss-Seidel iteration. Axis of symmetry points are treated as interior points by assuming a "mirror" grid opposite the axis of symmetry. Forcing functions are defined in order to force clustering of row of constant  $\eta$  toward the canopy surface boundary. In order to achieve the desired coordinate control, the following forcing functions are defined:

$$P=0.0 \quad (8)$$

$$Q=-Q_1 e^{-c(\eta-1)} \quad (9)$$

$Q_1$  is a positive constant and the magnitude of  $Q_1$  controls the intensity of the clustering towards  $\eta=\eta_1$ .  $c$  is a constant controlling the decay rate of clustering from  $\eta=\eta_1$ . For the special case in which there is no coordinate control and the forcing functions  $P$  and  $Q$  are zero, Poisson's equation reduces to Laplace's equation. Figures 6a-6c illustrate the process of generating a C-Grid about a parachute shape with the forcing functions given in equations (8) and (9). Figure 6a shows the initial algebraic grid surrounding the decelerator surface. The algebraic grid is created by defining the grid boundaries and distributing interior points linearly between the outer boundaries. The algebraic grid is used for an initial guess in the generation of the elliptic grid. Figures 6b-6c show the elliptic grid with no forcing functions (solution to Laplace's equation) and the elliptic grid with specified forcing functions ( $Q_1=500$ ,  $c=0.5$ ).

The grids in figure 6c appears to be well-behaved for the shown

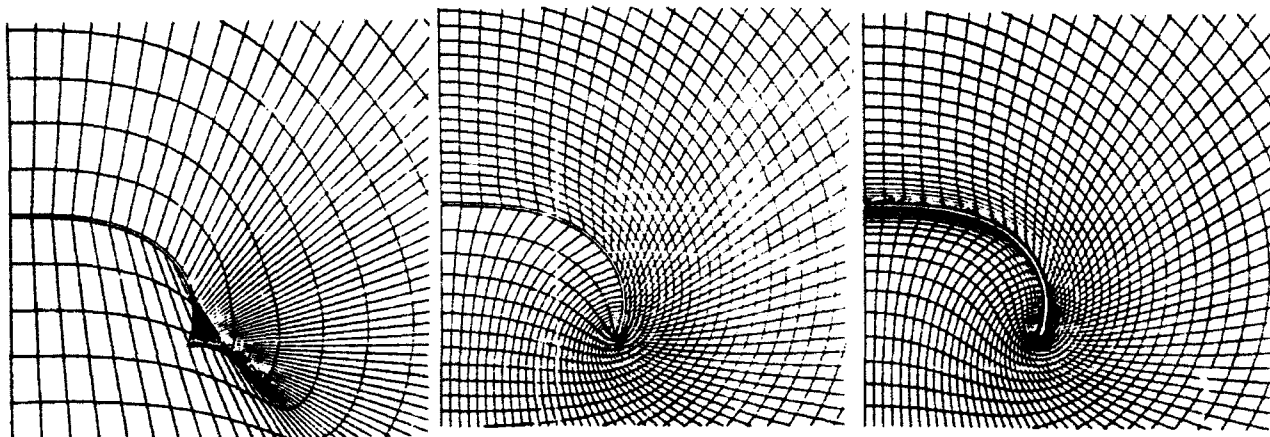


Figure 6. Generation of C-Grid

canopy shape. However, for canopies that are excessively concave (i.e. not nearly as fully opened), the gridding in the contained region of the canopy becomes very skewed and poorly behaved. Some "fine-tuning" was done in order to improve the gridding inside highly concave canopy shapes. The distribution of grid points below the canopy and along the symmetry axis were defined to improve the distribution of rows of constant  $\eta$  near the canopy surface ( $\eta=\eta_i$ ). For some computations it was beneficial to condense the distribution of outer surface grid points towards the symmetry boundary below the canopy surface and then to cluster rows of constant  $\xi$  towards  $\xi=\xi_{imax}$ . The desired clustering is achieved by defining the clustering parameter  $P$  as follows:

$$P = -P_1 e^{-d(\xi_{imax} - \xi)} \quad (10)$$

$P_1$  is a positive constant and the magnitude of  $P_1$  controls the intensity of the clustering towards  $\xi=\xi_{imax}$ .  $d$  is a constant controlling the decay rate of clustering from  $\xi=\xi_{imax}$ .

In contrast to the initial computational gridding strategy, the new gridding scheme uses different sets of points for defining the canopy surface in the structural and fluid dynamics codes. Thus, positions, velocities, and pressures along the surface must be interpolated in order to transfer information between structural and fluid dynamics' codes. The interpolation process will be described in greater detail later. Also, the canopy surface must

be given a finite thickness with the C-Grid scheme. This canopy surface thickness distribution is specified as constant everywhere except near the tip (or skirt). At the tip, the canopy surface is defined as a half circle with a diameter equal to the thickness of the canopy away from the tip. Each timestep the canopy surface positions, and exterior boundaries of the C-Grid are updated. Interior grid positions are then updated by solving Poisson's equation, with the same forcing functions used to generate the initial grid. Each time step, boundary conditions are imposed for the four outer boundaries of the computational grid. The condition imposed on the decelerator surface boundary is governed by the change in canopy nodal positions and the current timestep and represent a noslip surface. It should be noted that the surface nodes defining a canopy with thickness are different than the velocity of the corresponding canopy with no thickness. The canopy thickness imposes a rotational velocity component onto the canopy surface. For this reason, canopy surface velocity boundary conditions are not interpolated from the structural code values. Instead, canopy positions are interpolated from the structural code, the specified thickness is given to the updated canopy shape, and velocities are determined from the changes in position and the timestep. The outer boundary is considered the undisturbed, far field and both components of velocity are set to zero. The two boundaries along the symmetry axis are given free-slip conditions which require that there is no normal velocity component.

## Mass Spring Damper (MSD) Model

### Introduction to MSD model

The parachute consisting of canopy, lines and payload is modeled as a series of lumped mass points (nodes) connected by springs and dampers as shown in figure 7. The model is similar to the model reported in [12]. The MSD model fits into the coupled code as a set of Fortran subroutines. The MSD subroutines require a pressure distribution along the meridional length of the canopy and a time step as input. The program returns the position and velocity of each MSD node at the requested time. The MSD model is being developed as a separate set of subroutines so that other parachute models could be used in its place and/or it could be coupled with CFD codes other than SALE.

The MSD model is axisymmetric but includes some three dimensional considerations. The current model has been used to approximate flat circular solid cloth canopies such as a C-9. The ability to model other canopy types such as conical, and flat extended skirt have also been programmed and will be discussed. Newton's second law is applied at each user-defined node to obtain a set of coupled nonlinear differential equations. A free body diagram of a typical interior node on the canopy surface is shown in figure 8. The forces  $F_1$ ,  $F_2$ ,  $F_3$  and  $F_4$  applied to canopy node  $i$  are described below.

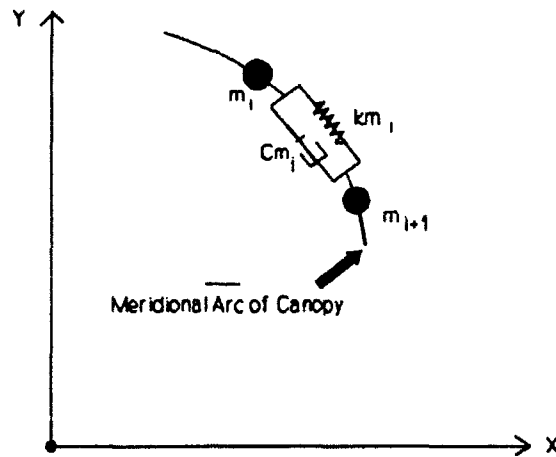


Figure 7. Mass Spring Damper Model

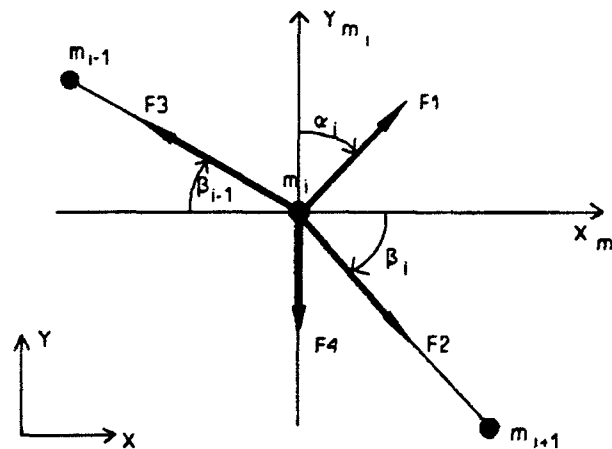


Figure 8. Free Body Diagram

•  $F1$  is the force due principally to the aerodynamic differential pressure acting across the canopy surface. The nodes are positioned along the canopy radials and the aerodynamic forces acting across the canopy are assumed to act at these nodes. Early versions of the model calculated the force  $F1$  at node  $i$  as the product of the current pressure difference over the canopy surface at node  $i$  and the current surface area associated with node  $i$ . The current model converts the pressure loading to localized forces at each node by using an approximation of the logic contained in the CALA code theory [13]. CALA determines the force per unit length applied to a radial by assuming a shape for the horizontal members. The horizontal members are infinitesimal strips of the gore connecting two radials. The CALA model assumes that the horizontal members take on a circular shape. The MSD model assumes that the pressure distribution across a horizontal section of the gore is constant for each time step. The CALA code requires a pressure distribution as input and predicts the "steady state" canopy shape

and stresses. The current MSD model does not include all of the theory in CALA. The model extracts the force per unit length at a given node based on the CALA geometric assumptions. The code then multiplies that force per unit length with the length associated with the given node. Special approximations are made to avoid singularities when the gore angle approaches zero by setting a minimum allowable gore angle. A special case is required for the node at the apex. This node's force is applied along the axis of symmetry and is equal to the current pressure difference at the apex node multiplied with the current surface area associated with the apex node. It should be noted that the direction of  $F1$  does not accurately represent the direction of the net force that the CALA logic calculates.

The force  $F1$  also includes a user defined variable viscous normal damping contribution. This damping is applied to the nodes based on the relative normal velocity of each canopy node to the payload velocity. The normal damping is included primarily to maintain numerical stability of the overall solution. The dampers are not attempting to model any physically observable phenomenon. The value used for the normal damping constant can have a major impact on the solution.

- $F2$  is the sum of the forces from the meridional spring and damper connecting nodes  $i$  and  $i+1$ . The spring force is the product of the spring constant and the change in length between nodes  $i$  and  $i+1$ . The spring force only acts when the distance between the nodes is greater than the constructed distance. The damping force opposes the relative velocity between nodes  $i$  and  $i+1$ . The force is the product of the damping constant and the relative change in velocity between nodes  $i$  and  $i+1$ . These dampers are included to damp out the high natural frequencies in the meridional springs. These natural frequencies may cause flow instabilities in connection with the "no-slip" boundary conditions at the canopy surface. There is also a force contribution from the CALA logic in the  $F2$  direction. This force and the CALA logic will be described in more detail later on in this report.

- $F3$  is the sum of the forces from the meridional spring and damper connecting nodes  $i$  and  $i-1$ . Similar to  $F2$ .

- $F4$  is the force due to gravity on node  $i$ . This force is the product of the gravitational acceleration constant and the mass of node  $i$ . The coupled model is solving the dimensional form of the governing equations. The coupled model could be used for similar simulations in other atmospheres with other gravitational fields.

The angles shown in figure 8 are described briefly below. There numerical approximations will be given later on in this report.

- $\alpha_i$  is the angle from the  $y$  axis to the outward normal direction associated with node  $i$ .

- $\beta_i$  is the angle from the local  $x$  axis of node  $i$  to the line segment connecting nodes  $i$  and  $i+1$ .



### MSD equations of motion

The equations of motion for the parachute are defined in this section. The canopy is modeled with a user defined number of nodes  $n$ . These nodes are defined along the radial with node 1 representing the apex and node  $n$  representing the skirt. The lines are also modeled with a user defined number of nodes  $nl$ . Line node # 1 is connected to the skirt and line node #  $nl$  is connected to the payload. The payload is defined by one node. The total number of nodes is equal to  $(n+nl+1)$ . The equations of motion for all canopy and line nodes have two degrees of freedom and are defined in the global X-Y coordinate system. The payload only has one degree of freedom in the global Y direction. Therefore the MSD model of the parachute is solving  $(2n+2nl+1)$  second order nonlinear ordinary differential equations (ODE's).

The equations of motion for all nodes are described in this section. The numerical implementation of various terms and the method used to determine the solution of these equations will be described in later sections. The interior canopy nodes are defined first. These include canopy nodes 2 through  $n-1$ . The acceleration in the x direction of canopy node  $i$  is given in equation (11).

$$\begin{aligned} \frac{d^2x(i)}{dt^2} = & \frac{1}{m_i} (CALAX + km_i \Delta l_i \cos \beta_i - km_{i-1} \Delta l_{i-1} \cos \beta_{i-1} - \\ & + Cm_i \frac{d(\Delta l_i)}{dt} \cos \beta_i - Cm_{i-1} \frac{d(\Delta l_{i-1})}{dt} \cos \beta_{i-1} - Cn_i [ \\ & \frac{dx(i)}{dt} \sin \alpha_i + (\frac{dy(i)}{dt} - \frac{dy(payload)}{dt}) \cos \alpha_i ] \sin \alpha_i ) \end{aligned} \quad (11)$$

The acceleration in the y direction of canopy node  $i$  is given in equation (12).

$$\begin{aligned} \frac{d^2y(i)}{dt^2} = & \frac{1}{m_i} (CALAY - km_i \Delta l_i \sin \beta_i + km_{i-1} \Delta l_{i-1} \sin \beta_{i-1} - \\ & Cm_i \frac{d(\Delta l_i)}{dt} \sin \beta_i + Cm_{i-1} \frac{d(\Delta l_{i-1})}{dt} \sin \beta_{i-1} - Cn_i [ \\ & \frac{dx(i)}{dt} \sin \alpha_i + (\frac{dy(i)}{dt} - \frac{dy(payload)}{dt}) \cos \alpha_i ] \cos \alpha_i ) - g \end{aligned} \quad (12)$$

Next, equations for a typical interior line node is given. These include line nodes # 2 through # nl-1. The acceleration in the x direction of line node i is given in equation (13).

$$\begin{aligned} \frac{d^2x(i)}{dt^2} = & \frac{1}{m_i} (km_i \Delta l_i \cos \beta_i - km_{i-1} \Delta l_{i-1} \cos \beta_{i-1} - \\ & + Cm_i \frac{d(\Delta l_i)}{dt} \cos \beta_i - Cm_{i-1} \frac{d(\Delta l_{i-1})}{dt} \cos \beta_{i-1} - DX_i) \end{aligned} \quad (13)$$

The acceleration in the y direction of line node i is given in equation (14).

$$\begin{aligned} \frac{d^2y(i)}{dt^2} = & \frac{1}{m_i} (-km_i \Delta l_i \sin \beta_i + km_{i-1} \Delta l_{i-1} \sin \beta_{i-1} - \\ & Cm_i \frac{d(\Delta l_i)}{dt} \sin \beta_i + Cm_{i-1} \frac{d(\Delta l_{i-1})}{dt} \sin \beta_{i-1} - DY_i) - g \end{aligned} \quad (14)$$

Next, equations of motion for special nodes are given. These nodes are node # 1 on the canopy, node # n on the canopy, node # 1 on the line, node # nl on the line and the payload node. The acceleration in the y direction of canopy node # 1 is given in equation (15).

$$\begin{aligned} \frac{d^2y(1)}{dt^2} = & \frac{1}{m_1} (F1 - km_1 \Delta l_1 \sin \beta_1 - Cm_1 \frac{d(\Delta l_1)}{dt} \sin \beta_1 \\ & - Cn_1 [ \frac{dy(1)}{dt} - \frac{dy(payload)}{dt} ] ) - g \end{aligned} \quad (15)$$

The acceleration of node # 1 on the canopy in the x direction is defined as zero due to symmetry if the node lies on the axis of symmetry (no vent). The acceleration in the y direction of the payload is defined next in equation (16). Note that the equation is for finite mass openings. The infinite mass opening payload equation will be described in a later section. The last node on the canopy is connected to the first line node with a line spring and a line damper. The last line node is connected to the payload with a line spring and a line damper. The equations for these special cases are simple modifications of the equations given above.

These equations are reformulated into a set of first order ordinary differential equations (ODE's). The ODE's are nonlinear in space and first order in time. The ODE's are solved over the desired time step with initial conditions by utilizing the SLATEC ODE solver DDEBDF and associated subroutines [11]. The subroutine

$$\frac{d^2y(\text{payload})}{dt^2} = \frac{1}{m_{\text{pay}}/N} (\Delta l_{n\text{line}} \sin \beta_{n\text{line}} k m_{n\text{line}} + C m_{n\text{line}} \frac{d(\Delta l_{n\text{line}})}{dt} \sin \beta_{n\text{line}}) - g \quad (16)$$

DDEBDF uses backwards differentiation formulas of orders one through five to integrate a system of first order ordinary differential equations. This formulation will be described in more detail later in this report.

#### CALA logic and equations

The MSD model is simply Newton's second law applied at every node representing the parachute. This requires a conversion of the pressure distribution along a radial that is supplied by the CFD code to be converted into a force vector at each node. A variety of techniques were used to accomplish this conversion.

The early MSD models assumed the canopy to be in the form of a surface of revolution and determined the magnitude of the force applied at each node as the product of the current surface area at each node and the supplied pressure distribution at that node. The force was assumed to act in the current normal direction of the surface at each node. The surface areas were approximated as conical sections between each node. This technique was refined by considering the pressure distribution to be piecewise linear and numerically integrating the product of the pressure distribution and the surface area for each node during each time step. One of the major problems with these models is that the models do not incorporate any type of hoop contributions. The nodes are located on the radials and the pressure distribution is acting across the gore surface. The model should account for the gore shape to apply loads to the radials even though the shape is axisymmetric in the CFD code. This approach should also incorporate a hoop component of force that earlier models considered only in the crudest sense. The desire to include a more accurate hoop force led to a crude version of the CALA theory being incorporated into the model.

The pressure distribution across the surface of the canopy is supplied by the CFD code at the MSD nodes. This is accomplished by normalizing the current radial arc length in both codes and using Lagrange Polynomials to interpolate values between the CFD and MSD canopy surface vertex and nodes. The MSD code utilizes the basic CALA assumptions to transform the pressure distribution into nodal forces along the radial. The MSD model nodes are located along a radial and the mass associated with each are lumped values based on the constructed shape of the canopy gores. The CALA code assumes that the horizontal members form sections of circular arcs and that the pressure distribution is uniform along the horizontals. The horizontal members lie in planes that are defined by the current

unit normal vectors from two adjacent radials making up one gore. The CALA reference defines the static force per unit radial length applied to a radial point as shown in equations (17) and (18) (Note these are equations (11) and (12) in the CALA reference [13]).

$$\frac{dF_{x_1}}{dl} = -\cos\psi \sin\frac{\pi}{N} \sin\theta \left(2 \frac{dH_{z_1'}}{dl}\right) \quad (17)$$

and,

$$\frac{dF_{y_1}}{dl} = \left( \frac{\sin\psi \cos(\pi/N)}{[1 - \sin^2\theta \sin^2(\pi/N)]^{0.5}} - \cos\psi \sin\frac{\pi}{N} \sin\theta \right) \left(2 \frac{dH_{z_1'}}{dl}\right) \quad (18)$$

Note that  $Y_1$  is defined as the local normal direction positive outward.  $X_1$  is defined as the local tangent direction with positive defined as pointing towards the skirt. Also the variable theta ( $\theta$ ) defining the normal direction in the CALA logic is related to the MSD variable alpha ( $\alpha$ ) by the equation  $\alpha = \pi/2 - \theta$ . Equations (17) and (18) give the force per unit length along the radial. The value of the horizontal force per unit length is given by equation (19) (Note this is equation (15) in the CALA reference).

$$\frac{dH_{z_1'}}{dl} = \frac{\Delta P y \sin(\pi/N)}{\sin\psi} [1 - \sin^2\theta \sin^2(\pi/N)]^{0.5} \quad (19)$$

The force per unit length equations in (17), (18), and (19) are converted to forces by multiplying each equation at each node by the corresponding current radial length associated with each node. The radial length associated with each node is taken as the one half the sum of the distances between the two nearest neighbor nodes. The variable  $\psi$  is the gore bulge angle that is determined iteratively at each node for each current surface configuration based on the constructed gore shape at each time step. The equation defining the gore bulge angle is equation (20) (Note this is equation (14) in the CALA reference).

In this equation  $N$  is the number of gores,  $y$  is the current distance from the node to the axis of symmetry and  $l_n$  is the constructed length of the horizontal member at that node. This equation is solved by Newton's method at each node and every time

$$\frac{\sin \psi}{\psi} = \frac{y \sin \pi / N}{l_h} \quad (20)$$

step. A few restrictions have been included to avoid singularities. The horizontal members load per unit length will blow up as the gore bulge angle approaches zero, see the denominator in equation (19). This is expected to occur near the apex of the canopy but could occur anywhere depending on the gore construction. The singularity is avoided in the code by defining a minimum allowable gore bulge angle. The force applied to each node are the forces per unit length in equations (17) and (18) multiplied by the current radial length associated with each node. Note that these forces include contributions in both the normal and tangential directions. A few extra assumptions are used in this dynamic model to include this crude version of the CALA logic. First, if there is no vent at the apex then the first node located on the axis of symmetry becomes a problem because the gore bulge angle is not defined at this node. The force at this node is taken as the product of the current surface area and the current pressure differential and is assumed to act along the axis of symmetry.

Another difficulty with this model is the "snap through" possibility. This will occur when the pressure distribution changes sign during a dynamic run. This can occur during a wake recontact near the apex and during the early stages of opening near the skirt. The model does not include any special logic for this possibility other than a change in sign for equation (18). Equation (17) is written in the code with an absolute value sign on the differential pressure term. Note that the current code does not attempt to include strain in the horizontal members even as the radial stretches. The horizontal members' length is considered fixed, based on the constructed gore geometry. This approximation can also lead to the singularities described above.

One last modification was included to deal with the early stages of the opening. The canopy is assumed to start with the gore horizontal members in sections of circular arcs as described above. There is a minimum distance from the axis of symmetry for each node at which contact between gores will begin. The minimum condition is based on simplified geometry considerations. The condition for contact to begin is given in equation (21).

$$y(i) < \frac{\cos(\pi/N) l_h(i)}{(\pi/2 + \pi/N) \sin(\pi/N)} \quad (21)$$

If contact exists, the code determines the distance of the contact region by assuming that the contact length is a straight line in the direction of the unit normal. The gore bulge radius is considered constant for a node with contact. The constant value is given in equation (22).

$$\psi = \pi/2 + \pi/N \quad (22)$$

The CALA logic described above is applied with the gore bulge angle constant and the new horizontal member length which is calculated as the original horizontal length minus the contact length. The forces determined are computed using the new horizontal length in equations (17) and (18). The tangential component is not modified due to the contact distance from the radial.

The CALA logic does take into account the gore geometry and allow the user to model a variety of gore shapes. The logic however has many assumptions that need to be addressed. The assumed shape and orientation being one of the major assumptions. Future coupled codes will have to address the first order three dimensional effects of the canopy. Future models will be discussed in a later section of this report.

#### MSD numerical approximations

This section describes the numerical approximations used in the code to determine the angle and distance terms in the equations of motion. The definition of  $\Delta l_i$  is given in equation (23).

$$\Delta l_i = [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]^{1/2} - l_{oi} \quad (23)$$

$$l_{oi} = [(x_{oi+1} - x_{oi})^2 + (y_{oi+1} - y_{oi})^2]^{1/2}$$

$\Delta l_i$  is the current change in length from node  $i$  to node  $i+1$ . The constructed length between nodes  $i$  and  $i+1$  is defined by  $l_{oi}$ . The first derivative of  $\Delta l_i$  with respect to time is the current change in velocity between nodes  $i$  and  $i+1$  and is given in equation (24).

The angular approximations are defined next and make reference to figure 9. The angle beta determines the current relative angle between the local x axis and the line segment connecting node  $i$  and  $i+1$ . The equations of motion require values of  $\sin(\beta_i)$  and  $\cos(\beta_i)$ . The  $\sin(\beta_i)$  is given in equation (25). The  $\cos(\beta_i)$  is given in equation (26).

The calculation for the angle alpha which defines the current

$$\frac{d\Delta l_i}{dt} = z l dx(i) * \left( \frac{dx_{i+1}}{dt} - \frac{dx_i}{dt} \right) + z l dy(i) * \left( \frac{dy_{i+1}}{dt} - \frac{dy_i}{dt} \right) \quad (24)$$

$$z l dx(i) = [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]^{-1/2} (x_{i+1} - x_i)$$

$$z l dy(i) = [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]^{-1/2} (y_{i+1} - y_i)$$

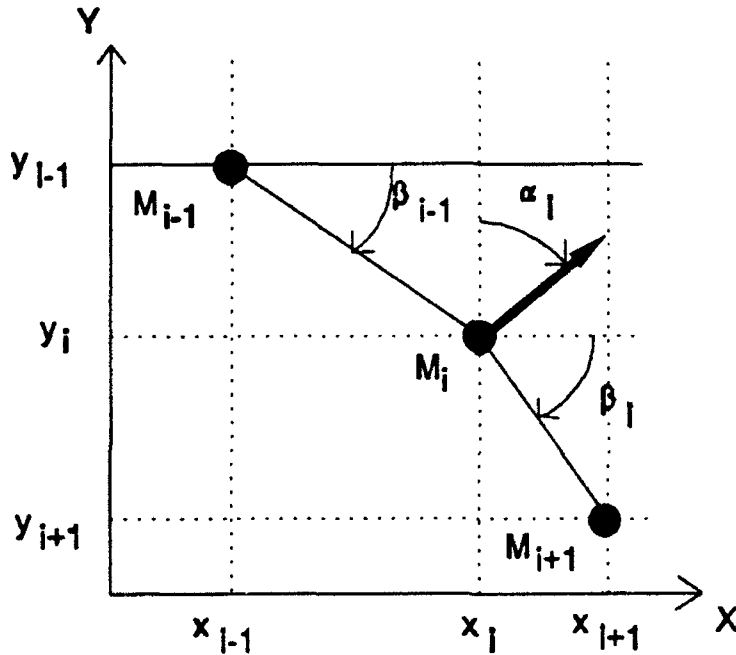


Figure 9. Definition of Angles

$$\sin \beta_i = \frac{y_i - y_{i+1}}{[(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]^{1/2}} \quad (25)$$

normal to the surface has taken on a variety of forms. The first method used in older models is given in equation (27) and would be accurate for equally spaced nodes. However, the spacing is not constant, even if the initial node layout is, due to stretching in the meridional direction.

This method was modified to adjust for unevenly spaced nodes by

$$\cos \beta_i = \frac{x_{i+1} - x_i}{[(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]^{1/2}} \quad (26)$$

$$\alpha_i = \frac{1}{2} (\beta_i + \beta_{i-1})$$

$$\sin \alpha_i = \frac{y_{i-1} - y_{i+1}}{[(x_{i+1} - x_{i-1})^2 + (y_{i+1} - y_{i-1})^2]^{1/2}} \quad (27)$$

$$\cos \alpha_i = \frac{x_{i+1} - x_{i-1}}{[(x_{i+1} - x_{i-1})^2 + (y_{i+1} - y_{i-1})^2]^{1/2}}$$

utilizing Lagrange polynomials. First, a third order Lagrange polynomial was fit through three consecutive nodes. The slope at the middle node was calculated and its negative inverse taken as the slope of the normal at the middle node. This method required special logic at areas where the tangent curve becomes infinite because the polynomial required to fit the points changes from a function of  $x$  to a function of  $y$ . The switching from a function of  $x$  to a function of  $y$  created a small discontinuity in the angle  $\alpha$  through time. The forces applied at the nodes are a function of  $\alpha$  and these small discontinuities caused local instabilities in the canopy motion which is believed to have caused small fluid instabilities.

The current model uses a weighted averaging technique which utilizes the current values of  $\sin(\beta_i)$  and  $\cos(\beta_i)$ . The values of  $\cos(\beta_{i-1})$  and  $\cos(\beta_i)$  are weighted to determine the value of  $\cos(\alpha_i)$ . The total current length of the two sections from node  $i-1$  to  $i+1$  is defined as  $L_i$ . Where  $L_i = l_{i-1} + l_i$ . The values of  $\sin(\alpha_i)$  and  $\cos(\alpha_i)$  are given in equations (28) and (29), respectively.

$$\sin(\alpha_i) = \frac{l_{i-1}}{L_i} \sin(\beta_{i-1}) + \frac{l_i}{L_i} \sin(\beta_i) \quad (28)$$

The surface area and meridional spring approximations will be discussed next. Two half gores and a radial are used to model the canopy as shown in figure 10. The number of nodes used to model the canopy (a total of  $n$ ) and the unstretched position of each node is user defined. The mass associated with each canopy node is



$$\cos(\alpha_j) = \frac{l_{j-1}}{L_j} \cos(\beta_{j-1}) + \frac{l_j}{L_j} \cos(\beta_j) \quad (29)$$

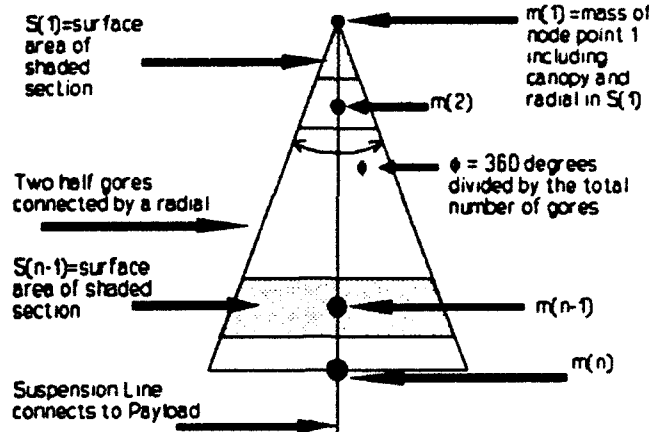


Figure 10. Layout for Two Half Gores

based on the undeformed geometry of the canopy. The undeformed surface area for a node is multiplied by the undeformed thickness of the canopy fabric and the density of the canopy fabric. This quantity is added to the mass associated with the radial contribution to that node.

The meridional springs are modeled by assuming a linear force versus deflection curve for both the fabric and the radial. The material associated with a meridional spring is considered as a rectangular section as shown in figure 11. The fabric spring and radial spring (optional line continuation) are considered to act in parallel. This approximation allows for the introduction of a Young's modulus term for both the canopy fabric and the radial. The equation to determine a meridional spring constant is shown in equation (30).

$$km_i = \frac{\pi E_2 h (x_{O_i} + x_{O_{i+1}})}{l_{O_i} N} + \frac{E_1 A}{l_{O_i}} \quad (30)$$

The apex and skirt nodes on the canopy require special treatment due to different surface area calculations.

The damping coefficient approximations will be discussed next. The damping coefficients in the MSD model equations of motion were usually taken as constants. The meridional damping constant  $Cm_i$  is given in equation (31) where  $Cm$  is a user defined value used for all meridional dampers.

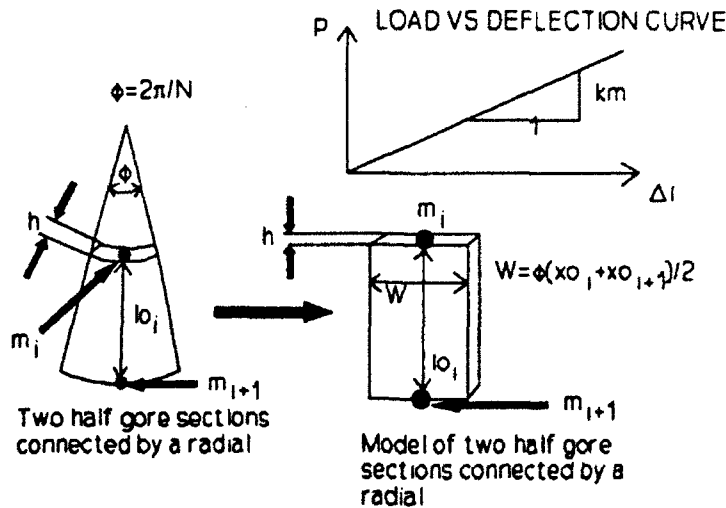


Figure 11. Meridional Spring Constants

$$Cm_i = 2 Cm (0.5 (m_i + m_{i+1}) km_i)^{0.5} \quad (31)$$

The normal damping constant  $Cn_i$  is given in equation (32) where  $Cn$  is a user defined value used for all normal dampers and  $m_{TOTAL}$  is the total mass of the canopy.

$$Cn_i = Cn (m_i / m_{TOTAL}) \quad (32)$$

The approximation used to model the normal drag contribution to the lines will be discussed next. The suspension lines have meridional dampers, but do not have the same type of normal damping as the canopy. The suspension lines normal motion is damped by incorporating a normal line drag approximation. The line drag approximation assumes that the velocity of the air is moving at the payload velocity and the model includes only the normal drag component to each line node. The approach is to first determine the current "outward" normal direction at each line node by fitting a third order Lagrange polynomial through each interior line node. The normal direction is defined by the angle alpha as with the canopy normal directions. The normal velocity component is calculated and based on each line nodes current velocity. Finally, the normal component of line drag is applied in the opposite direction of the current normal velocity for each line node. The

magnitude is  $D=C_D(\frac{1}{2}\rho[V_N]^2)S$  where,  $C_D$  is the drag coefficient [14] and approximately equal to (1.4) but can be changed through an input value.  $\rho$  is the density of the fluid and is assumed to be constant.  $V_N$  is the magnitude of the normal velocity at the line node.  $S$  is the normal component of area associated with each line node.  $S$  is computed as the product of the line diameter and the length of the line segment. The magnitude of  $V_N$  is given in equation (33).

$$|V_{Ni}| = \left| \frac{dx_i}{dt} \sin(\alpha_i) \right| + \left| \frac{dy_i}{dt} \cos(\alpha_i) \right| \quad (33)$$

The x and y components of line drag at each line node is determined and sent to the ODE solver as a constant for each time step. The sign of the x and y components of the drag are opposite the sign of the current x and y components of velocity respectively. The equations for the x and y components of the normal drag are given in equations (34) and (35).

$$DX_i = - \frac{dx_i/dt}{|dx_i/dt|} \sin(\alpha_i) C_D (\frac{1}{2} \rho [V_{Ni}]^2) S \quad (34)$$

$$DY_i = - \frac{dy_i/dt}{|dy_i/dt|} \cos(\alpha_i) C_D (\frac{1}{2} \rho [V_{Ni}]^2) S \quad (35)$$

#### Reformulation for SLATEC software

The parachute equations of motion consist of  $m=2n+2nl+1$  ordinary differential equations (ODE's). These coupled ODE's are second order in time. The coupled ODE's are reformulated into a larger set of coupled ordinary differential equations which are first order in time with a change of variables. This is accomplished by defining new variables for the derivatives with respect to time as shown in equation 36. This procedure has converted the  $m=2n+2nl+1$  ODE's into  $M=4n+4nl+2$  governing ODE's which need to be solved for each desired time step.

$$\begin{aligned} z_1 &= x & z_3 &= y \\ z_2 &= \frac{dx}{dt} & z_4 &= \frac{dy}{dt} \end{aligned} \quad (36)$$

$$\begin{aligned} \frac{dz_1}{dt} &= z_2 & \frac{dz_2}{dt} &= \frac{d^2x}{dt^2} \\ \frac{dz_3}{dt} &= z_4 & \frac{dz_4}{dt} &= \frac{d^2y}{dt^2} \end{aligned} \quad (37)$$

These ODE's are nonlinear in space and first order in time. The values of  $n$  and  $nl$  are user defined. However, changing the values of  $n$  and  $nl$  requires the program to be recompiled. The variables  $z_1, z_2, z_3,$  and  $z_4$  are vectors. Each element of the vector  $z_i$  ( $i=1-4$ ) has a unique value at each node on the parachute system at each time step. These values are the solution of the governing system of ODE's. This resulting system of equations can be written in an acceptable form for the SLATEC subroutine DDEBDF.f. The subroutine DDEBDF.f uses the backwards differentiation formulas of orders one through five to integrate a system of first order ordinary differential equations. The equations must be written in the format shown in equation 38. DDEBDF.f requires a separate subroutine be written which defines the differential equations. A set of initial conditions must also be specified.

$$\begin{aligned} \frac{DZ}{Dt} &= DF(t, Z) \\ \text{where } Z^T &= (Z_1, Z_2, Z_3, \dots, Z_M) \\ Z_1 &= (Z_1, Z_2, \dots, Z_{n+nl}), \quad Z_3 = (Z_m, Z_{m+1}, \dots, Z_{3n+3nl+1}) \end{aligned} \quad (38)$$

The solution of these dynamic equations also requires and depends on the boundary conditions and initial conditions applied to the parachute system. A variety of boundary conditions and initial condition options are included in the code as user defined options. These options will be discussed in the next section of this report.

#### Initial conditions

The MSD model includes a variety of initial condition options. The initial conditions required to solve the governing PDE's are to prescribe the initial position and velocity of every node in the MSD model. The model is restricted to starting with an initial shape that has a positive enclosed volume of sufficient size. This

restriction is due to the CFD's structured grid which must maintain a positive volume in its CFD cells for all time. Also, the MSD model is expected to be less accurate for these shapes which are less axisymmetric. A variety of initial shapes have been attempted. Most runs are initiated with the payload located at the origin of the global MSD coordinate system. The constructed dimensions of the canopy are known, so a constructed radial distance and a line length are known. The sum of the line length and radial arclength defines the total unstretched arclength of the parachute. Initial shapes were constructed by defining an angle from the axis of symmetry to the parachute suspension line (always assumed to be initially straight) and generating a conical base with a spherical section top where the total arc length is given by the constructed geometry. The angles used generally ranged from 3 to 5 degrees. Other initial shapes included a conical base and a conical top with the two cones connecting at the skirt.

The initial velocities of all nodes must also be specified to generate a solution. The simplest case is to set all velocities equal to zero. This is an approximate simulation of a canopy hanging from the apex with an initial volume defined above and no initial strain. This was the most common set of initial conditions used in an attempt to model ongoing experiments of free hanging parachutes.

Different initial conditions must be employed to more accurately model other types of real parachute openings. A variety of impulsively started runs were made with the payload given an initial velocity, but all other nodes and the entire fluid at rest. Also, a linearly impulsive initial set of velocities was attempted where the payload and apex node velocities are defined. All other nodes are assigned velocities linearly based on their global Y initial position of each node. These impulsive initial conditions are a first attempt at starting the numerical simulation of the parachute system at line stretch.

#### Boundary conditions

A variety of boundary conditions are included as user options in the model. The first option is skirt reefing. This is modeled by fixing the global X position of the skirt node. The node can be fixed in space or attached to a "nocompression" spring that is connected to the axis of symmetry at the same global Y coordinate as the skirt node for all time. This spring's stiffness could be based on the reefing lines characteristics. The restriction on the skirt node can be released at a user defined time, payload velocity or payload force. The logic to incorporate a second stage of skirt reefing would be a trivial modification in the MSD model.

A second boundary condition available in the code is all X reefing. This option applies the same type of restrictions as skirt reefing to all of the canopy nodes. This condition does not

restrict any vertical motion of the nodes. The "disreefing" of the nodes can be specified at a given time, payload speed, or payload force. This option was used to approximate infinite mass openings which are discussed in the next section. This type of boundary condition allows the parachute system to obtain a vertical velocity before opening. The model is not physical because the flow develops around and inside the canopy and therefore a relatively large pressure distribution is formed over the canopy surface before "disreefing" the global X displacement restriction.

The all X reefing option does have potential for other applications which have not been investigated at this time. For example, to determine "steady state" shapes, pressure and velocity fields. These runs involve starting with an initial shape that is close to the expected fully opened shape with zero velocity for all nodes. The run is marched out in time until the flow field is developed around the canopy. The X reefing restriction is then lifted to allow the canopy to move into its numerically predicted terminal shape. These same types of runs can be done with other initial shapes and with a large amount of damping at each node in the normal direction of the canopy surface. The normal damping option in the code is of a viscous type based on relative velocities of canopy nodes to the payload velocity or the system's center of mass. The damping parameters can be lowered in magnitude as a function of time when the parachute is approaching its steady state shape.

#### Infinite mass opening

An infinite mass opening is approximated in the model and is included as a user option. Choosing this option sets the gravitational constant to zero. The payload equations of motion are predefined smooth functions in time. The functions chosen for the acceleration and velocity of the payload are given in equations 39 and 40.

$$\frac{dy_{pay}}{dt} = \frac{v_{\infty}}{2} (1 - \tanh(v_{sp}(t - v_t)) - (1 - \tanh(-v_{sp}v_t))) \quad (39)$$

$$\frac{d^2y_{pay}}{dt^2} = -\frac{v_{\infty}}{2} v_t \left[ \frac{1}{\cosh(v_{sp}(t - v_t))} \right]^2 \quad (40)$$

Where  $v_t$  is the time at which the inflection occurs,  $v_{\infty}$  is the final payload velocity, and  $v_{sp}$  is a parameter used to define how fast the

payload velocity changes from zero to  $v_c$ . The velocity curves start from rest and accelerate to a near constant velocity value of  $v_c$  at the user defined time of  $2v_c$ . This option is run with the all X reefing option. The user defines the disreefing time which releases the X restrictions after the flow has developed at the terminal prescribed payload velocity. The canopy opens over time with the payload velocity remaining unchanged. This model does allow the pressure distribution to build up around the canopy before disreefing. It is expected that the influence and significance of this pressure buildup before disreefing will be minimized as the initial canopy shape becomes more tube-like with a smaller initial maximum diameter.

#### MSD FORTRAN program description

The solution of the equations of motion representing the parachute is performed numerically with a set of FORTRAN subroutines that are called from the CFD main program. The structural code is capable of solving the parachute problem for a variety of user supplied input parameters. This section gives a brief overview of the subroutines features and includes a summary flow chart which outlines the contents of the subroutine program.

The program has a variably spaced grid capability and a user-defined number of nodes option. The user defined number of nodes on the canopy ( $n$ ) and on the lines ( $nl$ ) are defined in a "parameter" statement in the subroutines. Therefore, the programs must be recompiled to change these options. The programs have been run with as few as 11 nodes and as many as 40 nodes to represent the canopy. Line nodes have ranged from zero for older versions of the code with no line nodes to 25 nodes. The variably spaced grid option allows the user to define the node locations on the unstretched canopy with unequal spacing. This option allows the user of the coupled codes to cluster nodes in areas of interest. The input files needed to run the coupled code are created with preprocessing programs.

The flow chart for the MSD FORTRAN subroutine programs is shown in figure 12. A description of each letter's role in the flow chart for the MSD code is given below.

A. ENTER MSD CODE: The CFD code provides the MSD code with the following information on each call. 1. The current pressure distribution on the canopy surface at each MSD node. 2. The current time. 3. The time at which a solution is desired (after initiation this is the current time plus the maximum prescribed time step). 4. Printing control parameters which tell the code whether or not and when to print a variety of output for postprocessing.

B. FIRST CALL ONLY: This section of the subroutine is for initialization and is only entered on the first call to the MSD

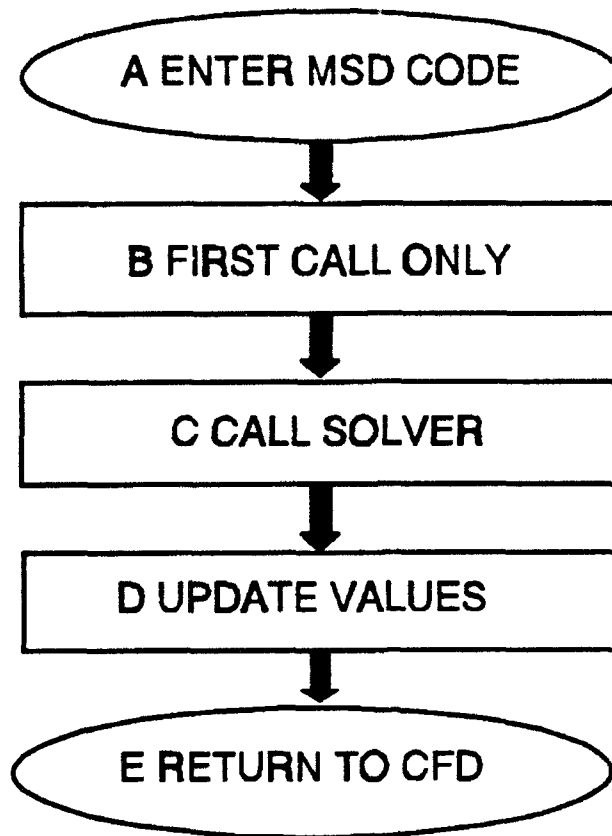


Figure 12. Flow Chart Outline for MSD Code Subroutine

subroutines. The MSD code opens an input file and reads in the following information.

1. Type of canopy.
2. Special information related to specific canopy, i.e. percent extended skirt, value of conical angle, etc.
3. Youngs modulus for the canopy fabric.
4. Canopy thickness.
5. Number of gores in the canopy.
6. Mass density of the canopy fabric.
7. Gravitational constant. If zero is entered, the code initializes an infinite mass opening and requires input of infinite mass opening parameters.
8. Payload weight.
9. Parameter (zero or one) which defines whether or not the lines continue through the radials.
10. Meridional damping constant.
11. Normal damping constant.
12. Minimum gore bulge angle allowed for the CALA logic.
13. Initial velocities of the payload and vent of the parachute



system. The initial velocity of all other nodes are linearly fit to these values based on the global Y coordinate system.

14. The next  $n$  lines are the initial X and Y node locations which are taken as unstretched in the meridional direction.

15. A parameter which tells the program to either start the run from the above unstretched initial shape or to read in an initial stretched shape from a separate file.

16. Youngs Modulus for the lines.

17. Mass density of the line material.

18. The next few entries define reefing information such as type of reefing, duration time of reefing etc.

19. The drag coefficient for the lines which is used in the program to determine the normal drag contribution to the line nodes.

This section of the code also determines the meridional spring constants, damping constants, node constant mass values (based on volume calculation times fabric mass density value), line node location (taken as evenly spaced if unstretched option is used otherwise they are read in), line spring values, line mass values, line damper values, gore horizontal dimensions, unstretched lengths for all springs, amount of initial stretch for all springs, relative velocities between connecting nodes, and initial values of the angles alpha and beta. This section also initializes a variety of parameters needed to utilize the SLATEC ODE solver, opens up output files, and writes initial condition information to these files.

C. CALL SOLVER: This section sets up values for and calls the SLATEC program DDEBDF.f which is linked to the correct set of governing equations that are based on the prescribed boundary conditions. The subroutine DDEBDF.f needs the following input, which is automatically determined by the FORTRAN program for each loop: 1. A subroutine name where the governing ordinary differential equations are located (The ODE's and subroutine must be written in a predescribed format). 2. Current deflections and velocities at each node. 3. Current time and desired output time. 4. A variety of inputs needed by the solver; for example the number of equations and the desired tolerances to be met. Before the call to the solver the code determines node forces due to the current pressure distribution based on the CALA logic previously described. The forces due to normal line drag are also determined. After the call to the SLATEC solver, the code updates the current spring elongations, current angle values, current relative velocity values, and current damping values.

D. UPDATE VALUES: This section extracts the desired output if requested for the current time step. This includes the current global node location, current displacements and velocities at every node, current pressure values at each node, hoop and meridional strain at each node and the hoop and meridional stress at each node. This section updates the current values for the next call and updates maximum and minimum values used for postprocessing.

The program writes to files that are set up in a MATLAB readable matrix format. These output files are read by the MATLAB software for postprocessing as discussed in the next section.

E. RETURN TO CFD: Return to CFD program with current positions and velocities of all nodes on the canopy and the position and velocity of the payload.

## Coupling

### Introduction

The coupling approach used in the model is an explicit marching method in time. The CFD code is used as the main FORTRAN program, which calls the structural code subroutines. The coupled model starts the computations with the flow medium and structural components at rest. The CFD solver computes the pressure distribution for the flow field at  $t=\Delta t$ , which is zero everywhere for the first time step for these initial conditions. The pressure distribution over the surface of the structural model and the time step are sent to the structural code. The structural code integrates the equations of motion over this time step at a user-defined set of nodes.

### CFD decelerator surface defined by interior grid points

For computations in which the SALE defines the decelerator with interior grid points, the coupling approach is straightforward, since each node in the structural representation on the canopy coincides with a specific adjacent CFD vertex. The positions and velocities of the canopy surface nodes are determined by the structural model and are returned to and updated in the CFD code. These surface vertices in the CFD code are required to move with the motion specified by the structural code. The boundary condition imposed by the CFD code on these vertices represents a no-slip boundary condition. The CFD code computes the pressure distribution for the next time step and sends the surface differential pressure values and the time step to the structural code. The process continues by marching forward in time up to a specified completion time. This process is described in greater detail in reference 5.

### Implementation of C-GRID

The fluid dynamics and structural dynamics were coupled for axisymmetric parachutes with a C-GRID. In coupled runs without the C-GRID, coupling between the structural and fluid codes was straightforward because the identical set of canopy surface nodes was used in both codes. With the C-GRID, coupling is not as straightforward because the structural and fluid codes use very different sets of nodes to define the canopy surface. A mesh update strategy and appropriate boundary conditions are needed in

order for the C-GRID surface boundary to be compatible with the structural representation of the canopy shape at each timestep.

The C-GRID for the initial canopy shape contains some information that is used throughout a coupled computation. The meridional distribution of nodes along the canopy surface and the canopy thickness distribution remain constant throughout the computation. Also, the clustering coefficients for the elliptic grid remain unchanged throughout the computation. For some computations it was necessary to keep the distribution of grid points below the canopy and on the symmetry axis constant throughout the computation. This was necessary to reduce the distortion of cells within the volume enclosed by the canopy surface.

The coupling process consists of three main steps which occur each timestep during the coupled solution. First, the C-Grid canopy surface is redefined each timestep in order to correspond to the current structural shape. Each timestep, the structural code returns updated positions and velocities for canopy nodes to the CFD code. These position are used to update the canopy surface in the C-Grid. Using the specified meridional distribution of points, the new surface points are determined by interpolation from the structural surface nodes using Lagrange polynomials. CFD surface points are interpolated from a fourth order Lagrange polynomial which is defined using the surrounding four structural surface nodes. CFD surface points near the axis of symmetry (between the first to structural surface nodes) are interpolated using a fifth order Lagrange polynomial defined using the first three structural surface nodes and assuming a "mirror" surface at the symmetry axis. The surface is then given a thickness normal to the interpolated positions as defined by the thickness distribution. (In some computations, the canopy is given a constant thickness. At the skirt, the surface is defined by a half-circle.)

Secondly, each timestep the elliptic grid is updated based on the updated canopy surface representation. The elliptic grid is defined by repositioning the surface boundary, repositioning the outer boundary, and then updating interior points and points along the symmetry axis. The outer boundary moves each timestep with the parachute payload. This keeps the canopy surface near the center of the grid throughout the computation. Interior grid points are updated with Poisson's equation and the same set of clustering coefficients that were used to generate the initial C-Grid. The C-Grid for the previous timestep is used as the initial "guess" for the updated C-Grid generation. Since the motion of the canopy surface is very small over a given timestep, the updated C-Grid converges after only a few iterations.

Finally, each timestep appropriate boundary conditions are defined. Boundary conditions on the canopy surface are defined to represent a noslip nonporous surface. Since the CFD canopy surface has a nonzero thickness, the surface velocities in the CFD representation are not the same as the surface velocities for the structural representation (which has no thickness). For this

reason, the velocity boundary condition for each surface point is simply taken as the displacement of that point over the previous timestep divided by the magnitude of the timestep. The outer boundary is assumed to be the far field and velocity components are defined as zero ( $u=v=0$ ) at the outer boundaries. The symmetry axis is given a freeslip boundary by setting the normal component of velocity ( $u$ ) to zero on the symmetry axis.

After the grid is updated and boundary conditions are specified as described above, the CFD solution is advanced one timestep. The resulting pressure distribution is then sent to the structural code as input. Again, this process is not as straight forward as for the initial gridding (see figure 4) approach since the CFD and structural codes use different sets of points to represent the canopy surface. This process consists of two main steps. First, the differential pressure distribution for C-Grid canopy surface is determined. Since SALE computes pressures at cell centers, nodal pressure values along the canopy surface are defined as the average of the pressures in the two surrounding cells. Pressures on the axis of symmetry nodes are defined as the pressure in the adjacent corner cell. Lower and upper surface pressures are subtracted to get the differential pressure distribution for the meridional distribution of surface points.

Secondly, the CFD pressure distribution is interpolated to the structural surface distribution. The pressure distribution is transferred from the CFD code to the structural code by the same approach that was used to transfer surface positions from the structural code to the CFD code. Using the distribution of nodes in the structural representation, the pressures are determined by interpolation from the CFD distribution of points using Lagrange polynomials.

Finally, the structural model uses the CFD pressure distribution to advance the structural solution one timestep and updated positions and velocities are once again returned to the CFD code. The C-Grid is then updated, boundary conditions are defined, and the process continues.

## MATLAB & AVS: Preprocessing & Postprocessing

### CFD preprocessing and postprocessing

The software packages MATLAB and AVS have been used as tools for visualizing CFD grids and flow fields. MATLAB and AVS are used in the generation of initial CFD grids for the coupled computations. MATLAB is used to determine the surface boundary points for the C-GRID and then to generate the initial algebraic grid. A FORTRAN program is used to generate the elliptic grid using the algebraic grid as an initial guess. Following the generation of the elliptic grid, AVS is used to visualize the grid in detail to determine if the grid is adequate for use in the computation. If the grid is not adequate, the surface boundary can be modified with MATLAB or attributes of the elliptic grid can be

modified with the FORTRAN program and then viewed again with AVS.

In order to view CFD flow fields, flow information is saved periodically throughout a computation for postprocessing. MATLAB and AVS serve as postprocessing tools. Several MATLAB scripts were written for quickly displaying CFD grids, velocity vectors, and pressure contour lines. AVS is used to look at CFD data in more detail and in color. With AVS, data can be viewed interactively and thus in a much more meaningful fashion than with MATLAB. AVS also has the ability to animate the computational results in order to view time-dependent flow field information.

#### MSD postprocessing

The MSD FORTRAN subroutines used to solve the equations of motion for the parachute system can generate a large quantity of numerical data. The number of dumps required and the time interval for these dumps are user defined input prior to the program execution. The data generated from a run must be analyzed in a logical and efficient manner. The MATLAB and AVS software have been used for postprocessing results from the MSD code. The AVS software is also used for viewing the axisymmetric deformations projected into 3-D space. The user must be able to extract information of interest including deformed shape, payload force, strains, and stresses all as a function of time. This requires the ability to graphically animate the motion and other information of the parachute system as a function of time.

MATLAB is capable of plotting a curve onto a fixed coordinate system. The data plotted can be read from any portion of a preloaded matrix. The curve can be "erased" by replotting it with the "invisible" option in MATLAB. Therefore, to create animation a curve is plotted then erased for one time step, then plotted and erased for the next time step, etc. The inclusion of a "pause" statement before the erasing of each curve allows the user to stop the animation at any time step. The MATLAB software is run on a Kubota mini supercomputer Titan 3000. The animation showing the results from a dynamic program run appears as uninterrupted motion to the human eye for runs with less than 30 nodes. A list of the output saved by the program in MATLAB matrix format and the capability of viewing various results with the MATLAB program are discussed below. The MATLAB programs are a modified version of programs written for a spherical membrane program that is discussed in [15]. This reference also includes the MATLAB source code for the spherical membrane model. A brief outline of the plotting sequence from the MATLAB program is given below.

#### MSD MATLAB program description

MATLAB first displays a listing of various input parameters of the run. Next, a list of available options appear which includes the key options given below.

1. The overall global shape of the canopy (as a two-dimensional meridional line figure) is animated in time in the global coordinate system. The canopy resolution may be small in this plot, depending on the distance the parachute system dropped during a run.
2. Same as 1 except viewed from a fixed payload perspective.
3. x or y position, velocity, acceleration, and pressure difference at any node versus time.
4. The pressure distribution across the canopy surface versus the canopy meridional length can be animated in time.
5. The hoop and meridional strains and stresses versus meridional canopy or line length can be animated in time.
6. The gore bulge angle versus meridional length can be animated in time.
7. Payload position and velocity or center of mass position and velocity versus time can be viewed.
8. Payload force versus time can be viewed.

A variety of other options are available. The program sets appropriate axes based on minimum and maximum values that are tracked during execution of the MSD coupled code or determined from the matrices with MATLAB.

#### AVS programs for MSD postprocessing

The AVS networks were written to generate a three-dimensional view of the parachute system. The data files for these programs are created by separate FORTRAN programs that read in the MATLAB data files and create the required AVS data files. One program utilizes the number of gores to generate that number of "rotated" data points of the canopy radials and lines. The user must define how many time steps are desired and the program generates the three-dimensional view based on radial positions and lines. The AVS software used for the simulation allows the user to view the opening in any orientation from a payload fixed reference point. A second FORTRAN program was written to extract the "steady state" three-dimensional shape in an AVS readable format. This program utilizes the CALA logic to determine three-dimensional data points on the canopy radials and across the horizontal sections of the gore based on the CALA logic. The result is a true three-dimensional view of the computed axisymmetric parachute system. The number of points to be calculated across the horizontals is a user defined parameter. The three-dimensional view can be rotated into any orientation within the graphics window. These views are used to compare computed terminal "steady state" shapes with real parachute shapes that are digitized either from video or still camera shots. The two images can be superimposed on top of each other to globally compare the computed and real shapes. This second technique will be expanded to allow for animation of the three-dimensional projection which will allow a side by side comparison of true airdrops to computed results.

## Results

The coupled computer model is being tested by modeling a variety of axisymmetric canopies. These canopies are either used by the U.S. Army for personnel/cargo or are scaled versions of common parachutes being used in experiments. Results from six simulations will be presented. Results from the first three simulations will be the basic information of the opening process. The fourth and fifth simulations were run to compare results to steady state permanently skirt reefed canopies. The last simulation will be presented in more detail. The six runs are summarized below.

1. A half scale C-9 dropped from rest modeling an experiment [16].
2. A quarter scale C-9 dropped from rest with the skirt reefed for one second and then allowed to disreef.
3. An infinite mass opening which is modeling an experiment [17]. The canopy is a small scaled flat circular that is opened in a wind tunnel. The speed of the air in the wind tunnel is maintained at 70 ft/sec.
4. A standard T-10 canopy (10 percent flat extended skirt  $D_o$  of 35 feet) is permanently skirt reefed. Terminal shapes of the canopy from the numerical simulation are compared to experiments [18].
5. A G-12 canopy (flat solid circular  $D_o$  of 64 feet) with a 3000 pound payload is permanently skirt reefed. Terminal shapes of the canopy and velocities from the numerical simulation are compared to experiments [19].
6. A half scale C-9 dropped from rest with all X reefing for 1.615 seconds. This run is modeling an experiment [16] in which the parachute free falls in a sleeve attached to a ceiling for 30 ft. The payload and lines are stored at the opening of the sleeve so the payload actually drops closer to 42 ft before the canopy begins to slide free from the sleeve.

Table 1 summarizes the input data used for each of the six cases.

TABLE 1. Summary of Input Parameters

PARAMETER DESCRIPTION	CASE 1	CASE 2	CASE 3	CASE 4	CASE 5	CASE 6
NX # of CFD Cells in the X Direction	70	59	29	70	70	70
NY # of CFD Cells in the Y Direction	60	104	49	60	60	60
n # of Nodes on the Canopy	24	29	11	25	25	24

nl # of Nodes on the Line	15	0	10	15	15	15
Payload Weight (pounds)	42.5	5.3	Inf. Mass	255	3000	51
Constructed Diameter (feet)	14	7	4.46	35	64	14
Constructed Line Length (feet)	12	5.74	3.66	29.4	51.2	14.2
Number of Gores	28	28	28	30	64	28

All of the cases except case 4 are modeling solid-cloth flat circular canopies with no porosity and no vent. All six cases were run with the same material properties listed below except where noted. The values used are very crude approximations of the experimental models because the goal of the model is to predict global results. The goal is to predict first order phenomena associated with parachute openings. The Young's modulus for the fabric and lines is taken as 4320000 (psf). The fabric thickness is 0.0004 (feet) and the line radius is 0.00593 (feet). The density of the canopy material is 0.593 (slugs/ft<sup>3</sup>) (1.2 slugs/ft<sup>3</sup> for case 5) and the density of the line material is 0.6 (slugs/ft<sup>3</sup>) (1.2 slugs/ft<sup>3</sup> for case 5). The gravitational constant is 32.2 (feet/second<sup>2</sup>) (case 3 does not include gravitational effects because the infinite mass tests were performed in a wind tunnel with the canopy opening horizontally to the ground). The fluid medium is air with standard atmospheric properties at sea level.

The initial shape of the canopies is unstretched. The initial shapes are determined by defining the angle between the y axis and the suspension lines. The canopy skirt is required to remain tangent to the suspension lines to create the initial configuration. The top of the canopy is required to trace out a circular section so that the required total line length and canopy constructed diameter are consistent the desired values. The initial payload position is defined as the origin of the y axis of symmetry.

The payload force in the payload force versus time curves for each case are calculated by taking the force in the suspension line spring that connects the last line node to the payload node and multiplying its vertical component by the total number of gores. The canopy skirt node is used for runs with know line nodes.

The six cases presented in this section were run on either a Kubota Titan 3000 mini supercomputer or on an Army Cray XMP Supercomputer. The maximum allowable time step varied for each run from  $4 \times 10^{-6}$  to  $5 \times 10^{-5}$  seconds. These time steps were required to maintain stability in the solution. The time steps are very small



and a simple calculation shows that a 5 second run, with even the larger time step, requires over 100,000 iterations. The coupled codes have not been optimized for performance. The goal has been to debug and get a feel for the effects of various parameters in the codes. Optimization and programming for specific high speed machine capabilities will be addressed for the next generation coupled codes.

The six cases will be described in more detail below. The figures associated with each case number are located in separate appendices.

### Case 1

Case 1 is a half-scale C-9 solid-cloth canopy, which is dropped from rest. This run is attempting to simulate ongoing experiments by Dr. Calvin Lee of Natick [16]. The parachute system is hanging from a ceiling with a release mechanism attached to the apex. The canopy hangs from the apex, the lines hang from the skirt and the payload is hanging from the bottom of the lines. The apex connection is released at time equal to zero seconds.

In the numerical simulation, the initial location of the payload node is at the origin. This simulation was repeated two times with different normal damping parameters. The simulation which resulted in the closest prediction of peak opening force will be shown in more detail. The other simulation and the effects of the normal damping parameter will also be discussed. Figure A1 shows a sequence of canopy shapes for equally spaced time steps from the initial shape at time equal to zero seconds up to time equal to 1.0 seconds. The figures for shapes at different times are presented from a fixed payload reference. Figures A2 and A3 are a continuation of figure A1 for times from 1.0 to 2.0 seconds and 2.0 to 3.0 seconds, respectively. The shape versus time curves show some of the first order characteristics that are typical with this type of parachute opening. These characteristics include the initial "ball" of air filling the apex of the canopy and then inflating the canopy from apex to skirt (see figures A1 and A2). The model also predicts a phenomena called wake recontact that occurs after the maximum payload force has been reached. Wake recontact occurs in relatively low payload mass to canopy surface area finite mass openings during or soon after the payload has undergone maximum deceleration. The wake trailing the opening canopy is moving close to the speed of the payload, so when the payload undergoes its maximum deceleration (often when the canopy is approaching the inflated diameter) the parachute system's vertical speed slows down and the trailing wake contacts the apex of the canopy. The recontacting wake can indent the apex of the canopy. This can be seen in figures A2 and A3.

The predicted payload force, velocity and position curves as

functions of time are plotted together in figure A4. Figure A5 plots the payload force versus time curve for the numerical simulation and two sets of payload force versus time curves from the experiments [16]. Even with the assumed initial shape and other approximations used in the model, the force versus time curves do show similarities. The peak force prediction from the model is strongly dependent on the normal damping parameter. For low values of normal damping the model predicts considerably higher peak loads than the experiments. The discrepancy in the peak force value is most likely the result of contributions from many factors. These factors include the axisymmetric assumption, no-porosity in the model, relatively crude structural model and the constants used for the material properties. The numerical model does predict the time at which the peak opening force will occur. The model also appears to predict the second peak in the payload force versus time curve which corresponds to a time after the wake recontact phenomena has occurred. The influence of the normal damping parameter on this simulation will be presented next.

As the normal damping constant decreases, the predicted peak opening force increases. Very large normal damping constants also effect the amount of time required for opening. The simulation was run with two different normal damping constants. Figure A6 plots the force versus time curves for the two simulations. Figure A6 shows the difference in the peak force values and the difference in the amount of fluctuations in the payload force. Figure A7 plots the payload velocities versus time curves. The velocity versus time curves also show the effects of the normal damping parameter on the solution. Figure A8 shows the payload position versus time curves for each simulation. The effect of the normal damping parameter on position is not significant as expected. The ability to predict the peak opening force without a normal damping contribution in the model is a goal for the next generation of the coupled codes.

## Case 2

A quarter scale C-9 is dropped from rest with the skirt reefed for one second and then allowed to disreef. This simulation was not based on an experiment. The run was made on an older version of the code for comparison with the results presented in [5] for which there was know reefing. The skirt reefing is accomplished by restricting global X displacements of the canopies skirt node for one second. At one second the restriction imposed on the skirt node is removed.

In the numerical simulation, the initial location of the payload node is at the origin. Figure B1 shows a sequence of canopy shapes for equally spaced time steps from the initial shape at time equal to zero seconds up to time equal to 1.0 seconds. The figures for shapes at different times are presented from a fixed payload reference. Figures B2 and B3 are a continuation of figure

B1 for times from 1.0 to 1.3 seconds and 1.3 to 2.3 seconds, respectively. The time interval from 1.0 to 1.3 shows the disreefing progression. The final shape and flow field is approaching the results reported in [5].

The predicted payload force, velocity and position curves as functions of time are plotted together in figure B4. The peak force predicted is greater than that predicted in [5] because the parachute system in this case reaches a greater velocity before disreefing.

### Case 3

This case is modeling a solid flat circular canopy under infinite mass opening conditions and was compared to experimental results presented in [17]. The canopy is a scaled C-9 flat circular that is opened in a wind tunnel with a maintained air speed of 70 ft/s. The numerical model simulates the infinite mass condition by prescribing a payload velocity versus time curve where the payload velocity approaches a terminal velocity.

In the numerical simulation, the initial location of the payload node is at the origin. Figure C1 shows a sequence of canopy shapes for equally spaced time steps from the initial shape just before disreefing at time equal to 3.0 seconds up to time equal to 3.1 seconds. Note that the initial shape shown in C1 is actually stretched in the global Y direction from the shape at time equal to zero (refer to the infinite mass subsection of this report for more details on infinite mass openings). The figures for shapes at different times are presented from a fixed payload reference. Figures C2 and C3 are a continuation of figure C1 for times from 3.1 to 3.2 seconds and 3.2 to 3.3 seconds, respectively. The shape versus time curves show some of the first order characteristics that are typical with infinite mass openings. The model does not predict the wake recontact phenomena. This is expected because the payload never decelerates in an infinite mass opening.

The predicted payload force and velocity curves as functions of time are plotted together in figure C4. The payload velocity curve was predefined. The curve shows the payload velocity starting at zero. The payload is smoothly accelerated and decelerated until the terminal (desired wind tunnel speed) is reached. In figure C4, the payload approaches 70 ft/sec at 1.7 seconds. The flow field is permitted to develop for another 1.3 seconds. At 3 seconds the reefing restriction on all of the canopy nodes is removed. Figure C5 plots the maximum diameter over the constructed diameter versus time. This curve was compared to the experimental curves on pages 16,17 and 18 of [17]. The numerical predictions occur slightly sooner in time and the over inflation (peak diameter reached before steady state) is lower in magnitude than the experimental results shown in [17]. The difference in the

time of overinflation is most likely due to the initial conditions imposed in the infinite mass opening. The numerical model has a positive volume inside the canopy and a significantly larger initial diameter. The flow field is permitted to approach steady state conditions and a significant pressure builds up inside the canopy before disreefing. The canopy in the experiment is opened from a more streamlined tube like shape. The discrepancy in the value of the over inflation may be because the numerical model is actually plotting maximum diameters of the radials position while the experiments are reporting maximum diameters based on the inflated gore shapes detected from video images of the experiments.

#### Case 4

A standard T-10 canopy (10 percent flat extended skirt  $D_0$  of 35 feet) is permanently skirt-reefed. Terminal shapes of the canopy from the numerical simulation are compared to experiments [18]. In this case the skirt reefing is accomplished by restricting global X displacements of the canopies skirt node after the desired skirt diameter is reached during the inflation. The initial skirt diameter was smaller than the desired reefed skirt diameter. This case and case 5 are presented to show other potential applications of future and more accurate coupled codes.

In the numerical simulation, the initial location of the payload node is at the origin. Figure D1 shows a sequence of canopy shapes for equally spaced time steps from the initial shape at time equal to zero seconds up to the final time equal to 5.0 seconds. The figure of shapes at different times is presented from a fixed payload reference. The figure shows the skirt initially moving inward towards the axis of symmetry, but then inflating and being restricted to the final prescribed skirt diameter.

The predicted payload force, velocity and position curves as functions of time are plotted together in figure D2. These are presented for clarity. The simulation was run to determine a steady state reefed shape and to estimate the terminal velocity of the parachute system. The steady state shape for the reefed T-10 canopy with a 255-pound payload is shown side by side with a picture of the actual drop in figure D3. The numerically predicted shape is on the left and the experimental image is on the right of figure D3. The drops were videotaped on standard VHS tapes [18]. The authors viewed the tapes and scanned in individual frames that appeared to be the most "axisymmetric." The reefed T-10 never reaches a truly steady state (nor do any real canopies) but the frame presented in figure D3 was viewed as typical of the drops. The frame was digitized and has been dithered and grayscaled for this report. The numerical image was created with the AVS software as described in a previous section of this report. The image was scaled and manipulated into approximately the same orientation as the experimental image to allow for a global shape comparison. It

should be noted that the experimental canopy has a vent at the apex. The experimental image shows an inward gore shape near the skirt which corresponds to a net inward pressure near the skirt. The net pressure for the numerical model is outward from apex to skirt.

The use of the code to predict terminal shapes and terminal velocities is relatively inefficient. However a sequence of runs could be made with different skirt reefing diameters and different payload velocities. The authors are attempting to couple a static structural code to a CFD code producing steady state results about fixed shapes. The codes would be utilized with an iterative process to converge on the steady state shape and flow field for a given Reynolds number. The payload weight would be part of the output.

#### Case 5

A G-12 canopy (flat solid circular  $D_0$  of 64 feet) with a 3000 pound payload is permanently skirt reefed. Terminal shapes of the canopy and terminal velocities from the numerical simulation are compared to experiments [19]. This case is similar to case 4 except the initial grids were created about the reefed skirt diameter. Therefore the skirt canopy node is restricted from any global X displacements for all time. This problem is presented because the canopy and payload are significantly larger than the other simulations presented. The reefed G-12 canopy was recently devised and approved as an alternate parachute for Operation Provide Promise. Operation Provide Promise includes dropping of relief supplies into Bosnia-Herzegovina.

In the numerical simulation, the initial location of the payload node is at the origin. Figure E1 shows a sequence of canopy shapes for equally spaced time steps from the initial shape at time equal to zero seconds up to the final time equal to 10.0 seconds. The figure of shapes at different times is presented from a fixed payload reference.

The predicted payload force, velocity and position curves as functions of time are plotted together in figure E2. These are presented for clarity. The simulation was run to determine a steady state reefed shape and to estimate the terminal velocity of the parachute system. The steady state shape for the reefed G-12 canopy with a 3000-pound payload is shown side by side with a picture of the actual drop in figure E3. The numerically predicted shape is on the left and the experimental image is on the right of figure E3. The image manipulation to produce figure E3 was described in case 4. The numerically predicted terminal velocity is approaching 70 ft/sec in figure E2. The experimental drops were analyzed and a terminal decent rate of 75.2 ft/sec was determined as an average for drops conducted from 15,000 ft. The terminal velocity results appear to be very close considering the

approximations used in the model. Other terminal descent rates and shapes will need to be compared for future models. Any numerical model that is predicting the opening behavior of a canopy should at the very least be capable of predicting the steady state characteristics of that parachute system.

#### Case 6

Case 6 is a half-scale C-9 dropped from rest with all X reefing for 1.615 seconds. This run is modeling an experiment [16] in which the parachute free-falls in a sleeve attached to a ceiling for 30 ft. The lines are stored at the opening of the sleeve so the payload actually drops closer to 42 ft before the canopy slides free from the sleeve. The sleeve is conical in construction. The canopy is not folded along the radials in the sleeve.

Figure F1 shows a sequence of canopy shapes for equally spaced time steps from close to the initial shape at time equal to 1.6 seconds just before disreefing up to time equal to 2.0 seconds. Figures F2, F3 and F4 are continuations of figure F1 for times from 2.0 to 2.5 seconds, 2.5 to 3.5 seconds and 3.5 to 5.0 seconds, respectively.

The predicted payload force, velocity and position curves as functions of time are plotted together in figure F5. The force is calculated by taking the force in the suspension line spring that connects the last line node to the payload node and multiplying its vertical component by the total number of gores. Figure F6 plots the payload force versus time curve for the numerical simulation and the payload force versus time curve from the experiment [16]. The two curves in figure F6 show similarities, but appear to be offset by approximately 0.4 seconds. The major reason for this offset is due to the required initial shape in the numerical model. As discussed previously in this report, the codes are currently limited to initial conditions that have a positive volume inside the canopy. This positive volume and the initial predefined skirt diameter change very little for the first 1.615 seconds of all X reefing. Therefore, during the first 42 feet of freefall the pressure builds up on the inside of the canopy. At disreefing, a substantial pressure has built up on the inside of the canopy which aids in the inflation process. This pressure build up initiates the inflation faster than the experiment.

Figure F7 plots the payload velocity versus time curve for the numerical simulation and the velocity versus time curve from the experiment [16]. The velocities from the experiments were obtained from an image analysis of the parachute system [16]. The curves have the same offset described above. Figure F8 plots the skirt and maximum diameter versus time curves for the numerical simulation and the skirt and maximum diameters from the experiment [16]. The experimental diameters were extracted from video that was shot from the payload looking "up" the lines at the canopy.

The same time offset is present.

A sequence of pressure and velocity fields for the volume surrounding the canopy are shown in figures F9 through F16. These plots are snapshots in time of the flow field surrounding the canopy. The snapshots are zoomed-in pictures of the flow field around the canopy and do not show the entire computational region. The pressure contour lines are shown on the left hand side of the figure. A description of the contour lines values accompanies each figure. The pressure range between contour lines' in these figures is 0.25 psf. The ambient pressure is shown on the figures as a bold contour line. The right hand side of these figures show the velocity vectors. The velocity vectors are scaled equally for each snapshot to provide information on the time-dependent velocity field in a consistent manner.

Figures F17 through F20 are a sequence of the computational CFD grids corresponding in time to each of figures F9 through F16. Each of figures F17 through F20 has two grids at different times. The grids progress in time from left to right. The difficulty of creating a structured grid which can follow and remesh around all of the potential canopy shapes is evident from this sequence of grid snapshots. CFD codes that utilize unstructured grids are being investigated for next generation CFD codes to help eliminate this difficulty. Unstructured grids will also provide more flexibility on the initial shapes of the canopy.

### Discussion

#### Computational Fluid Dynamics

The CFD model has been able to represent canopy geometries undergoing large deformations during the opening process. Although many modifications have been made to adapt codes to study the opening process, the current model still has several limitations to be addressed in the future. Inclusion of fabric porosity into the model should reduce the predicted opening shock and improve the correlation between predicted and experimentally measured behaviors. It is also believed that incorporation of a turbulence model will improve the correlation of predicted behavior with experimental data. One of the primary limitations of the current model is its inability to represent canopy geometries early in the inflation phase due to gridding restrictions. Future models will need to upgrade the current gridding approach. Utilization of unstructured grid CFD codes with periodic remeshing may be the best approach.

#### Mass Spring Damper Model

The MSD model has many assumptions and is not expected to model axisymmetric canopies completely. However, the model is capable of modeling large deformations that are similar to those

experienced by parachutes. At this time we believe the MSD model has reached its potential for predicting parachute openings. The experience, techniques learned, and some sections of the codes will be very useful for potential next-generation models. A higher-order model should more accurately predict structural behavior for parachute systems. Presently, four separate source codes are being considered and evaluated. Each of these codes includes a three-dimensional capability and all are based on the finite element method.

### Coupling

The current model could continue to undergo improvement. The explicit marching method should approach the exact solution to the coupled equations for a sufficiently small time step. However, an implicit method, which would require iterating between the structural and CFD codes, may be advantageous in the future. A variety of next generation codes are being reviewed for their potential use in future generation coupled codes. The ability to efficiently predict terminal "steady state" information for any type of axisymmetric parachute system is also being investigated. The codes could allow a user to make a variety of small modifications to a design and compare the steady-state results such as terminal drag, velocity and shape in an acceptable quantity of time. This would provide the user accurate terminal decent information for drops in which the opening is not of primary concern and thus eliminate the need for running a more costly dynamic code.

### Conclusions

The complexity of modeling the opening process stems from the coupling between the structural dynamics of the canopy, lines plus payload and the aerodynamics of the surrounding fluid medium. This paper has described ongoing research at Natick which involves the coupling of a CFD code and a structural dynamics code. The solution to the coupled problem is expected to assist in the development of future U.S. Army airdrop systems, which include the capability of deploying at low altitudes and high speeds. Initial computational results with the model described in this paper compare favorably with experimental data. However, the current model requires significant improvements and enhancements before it can be considered usable as a design aid. However, estimates of some design parameters can be made with the present model. Future computational models are expected to provide significant insight about the behavior of parachutes during the opening process.



## REFERENCES

1. Steeves, Earl C. "Prediction of Decelerator Behavior Using Computational Fluid Dynamics." *Proceedings of AIAA 9th Aerodynamic Decelerator and Balloon Technology Conference*. Albuquerque, NM. October 7-9, 1986.
2. Steeves, Earl C. "Analysis of Decelerators in Motion Using Computational Fluid Dynamics." *Proceedings of AIAA 10th Aerodynamic Decelerator and Balloon Technology Conference*. Cocoa Beach, FL. April 18-20, 1989.
3. Stein, Keith R. "Computations of the Flow Characteristics of Aerodynamic Decelerators Using Computational Fluid Dynamics." *Proceedings of AIAA 11th Aerodynamic Decelerator and Balloon Technology Conference*. San Diego, CA. April 9-11, 1991.
4. Stein, Keith R. "An Investigation of Parachute Aerodynamic Characteristics Using Computational Fluid Dynamics." *Proceedings of the 4th Natick Science Symposium*. Natick, MA. June 9-10, 1992.
5. Stein, Keith R., Benney, Richard J., and Steeves, Earl C. "A Computational Model that Couples Aerodynamic and Structural Dynamic Behavior of Parachutes During the Opening Process." U.S. Army Natick Research, Development and Engineering Center. Natick Technical Report No. NATICK/TR-93/029. April 1993.
6. Peterson, Carl W. "The Fluid Physics of Parachute Inflation." *Physics Today*. August 1993.
7. Amsden, A.A., Ruppel, H.M., and Hirt, C.W. "SALE: A Simplified ALE Computer Program for Fluid Flow at All Speeds." Los Alamos Scientific Laboratory Report No. LA-8095, 1980.
8. Hoffman, Klaus A. "Computational Fluid Dynamics for Engineers." Education System, Austin, Texas. 1989.
9. Steger, J.L. and Sorenson, R.L. "Automatic Mesh-Point Clustering Near a Boundary in Grid Generation with Elliptic Partial Differential Equations." *Journal of Computational Physics*, Volume 33. 1979.
10. Thompson, J.F., Warsi, Z.U.A., and Mastin, C.W. "Numerical Grid Generation Foundation and Applications." North-Holland. 1985.
11. SLATEC Library, A collection of FORTRAN mathematical subprograms available through the National Energy Software Center (NESC)

12. Sundberg, W.D. "Finite-Element Modeling of Parachute Deployment and Inflation." *Proceedings of AIAA 5th Aerodynamic Deceleration Systems Conference*. Albuquerque, NM. November 17-19, 1975.
13. Sundberg, W.D. "New Solution Method for Steady-State Canopy Structural Loads." *Journal of Aircraft*. Vol. 25, No. 11. November 1988.
14. Hoerner, Sighard F. "Fluid Dynamic Drag." Published by the Author, 1965.
15. Benney, Richard J. "A Nonlinear Dynamic Spherical Membrane Model." U.S. Army Natick Research, Development and Engineering Center. Natick Technical Report No. NATICK/TR-93/017. January 1993.
16. Lee, C.K. Private Communications. U.S. Army Natick Research, Development and Engineering Center.
17. Melzig, H.D., Schmidt, P.K. "Pressure Distribution During Parachute Opening. Phase 1. Infinite Mass Operating Case." Air Force Flight Dynamics Laboratory, Research and Technology Division, Air Force Command, Wright-Patterson Air Force Base, OH. Technical Report AFFDL-TR-66-10, March 1966.
18. Giebutowski, E. Private Communications. U.S. Army Natick Research, Development and Engineering Center.
19. Watkins, J. Private Communications. U.S. Army Natick Research, Development and Engineering Center.

BLANK PAGE

## **APPENDICES**

BLANK PAGE

**APPENDIX A**

**CASE 1**

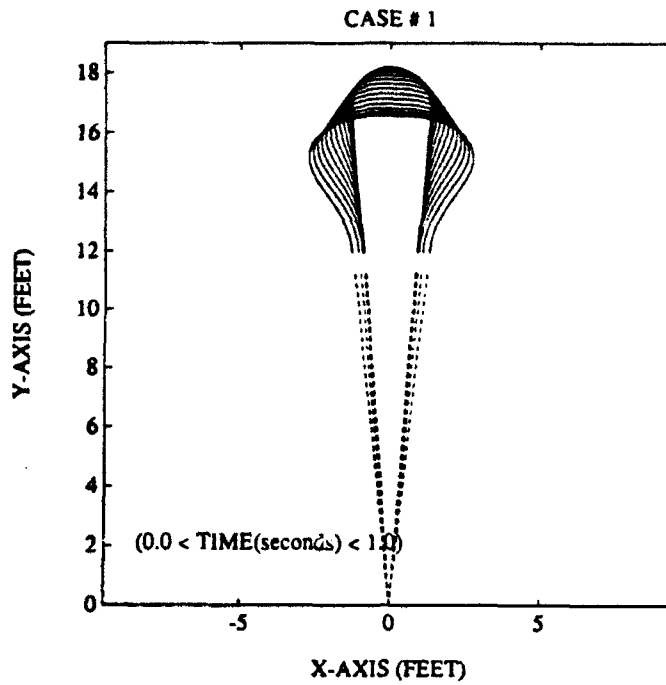


Figure A1. Canopy Shape Versus Time in Seconds ( $0.0 < t < 1.0$ )

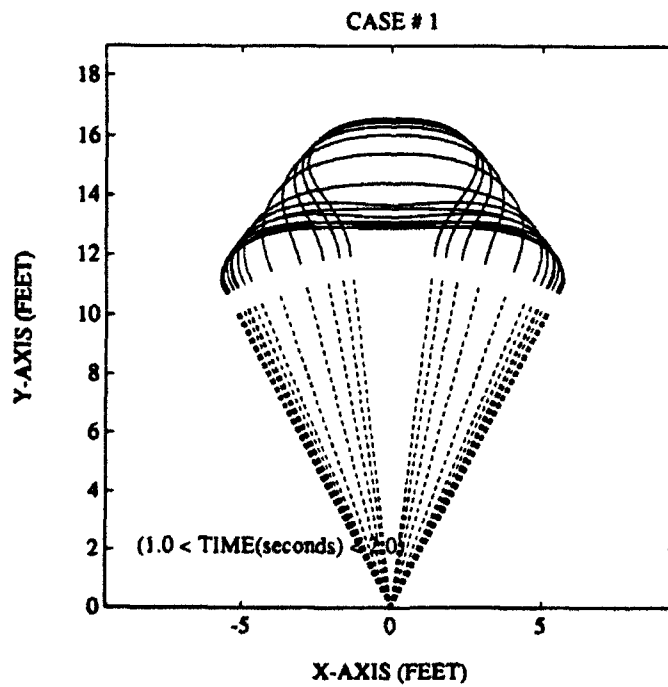


Figure A2. Canopy Shape Versus Time in Seconds ( $1.0 < t < 2.0$ )

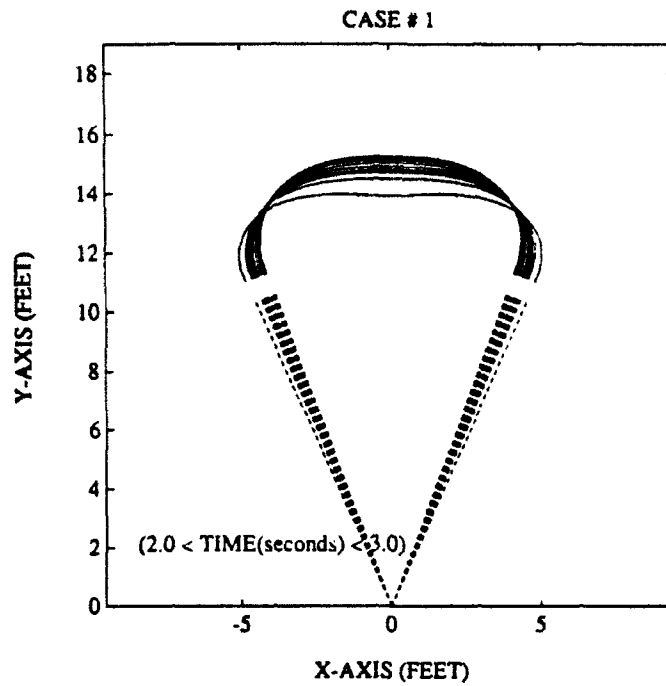


Figure A3. Canopy Shape Versus Time in Seconds ( $2.0 < t < 3.0$ )

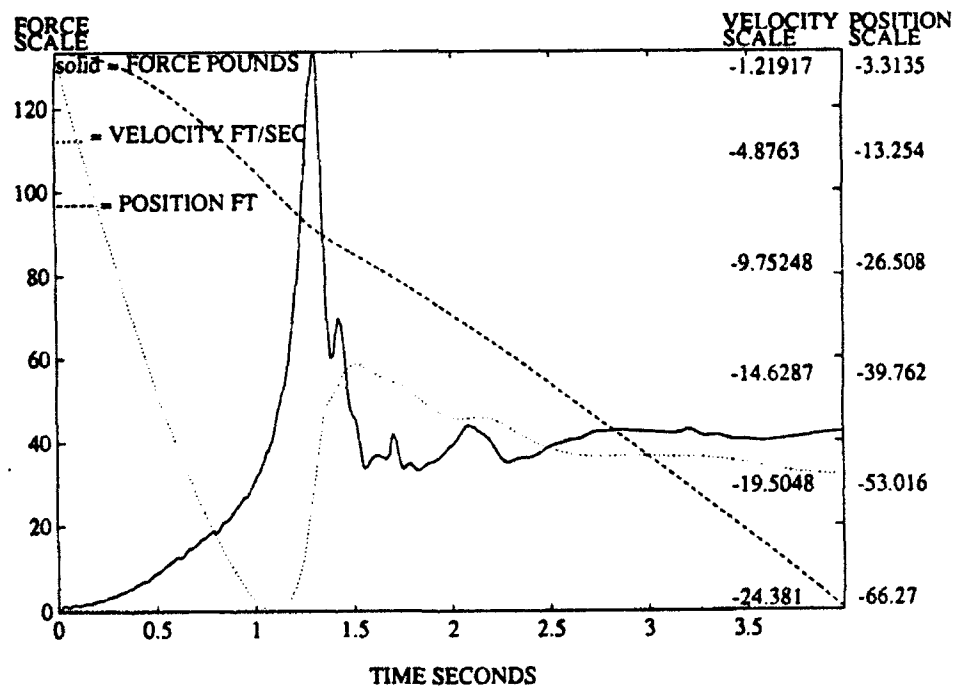


Figure A4. Payload Force, Velocity & Position Versus Time



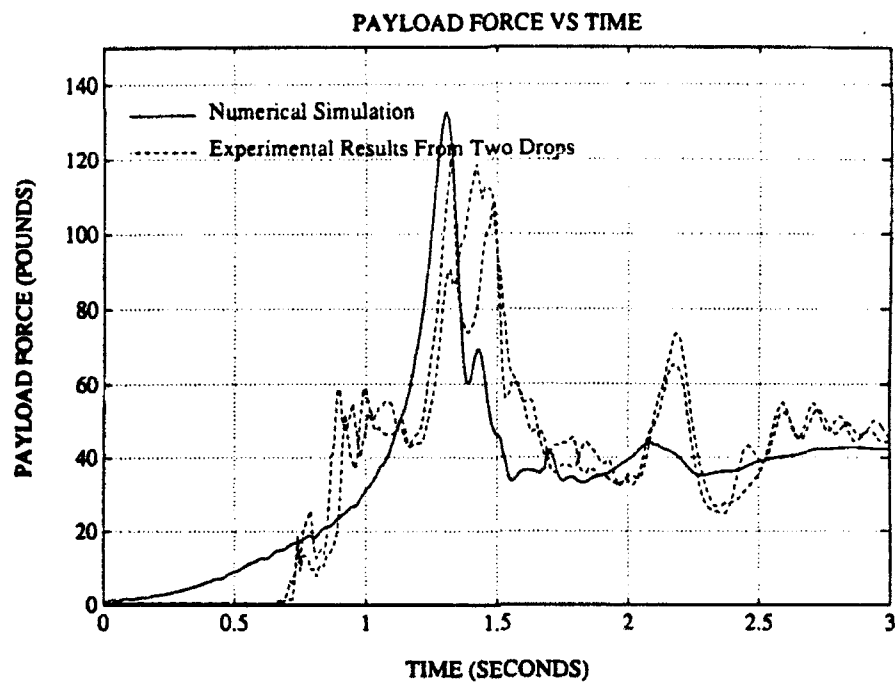


Figure A5. Payload Force Versus Time (Numerical & Experimental)

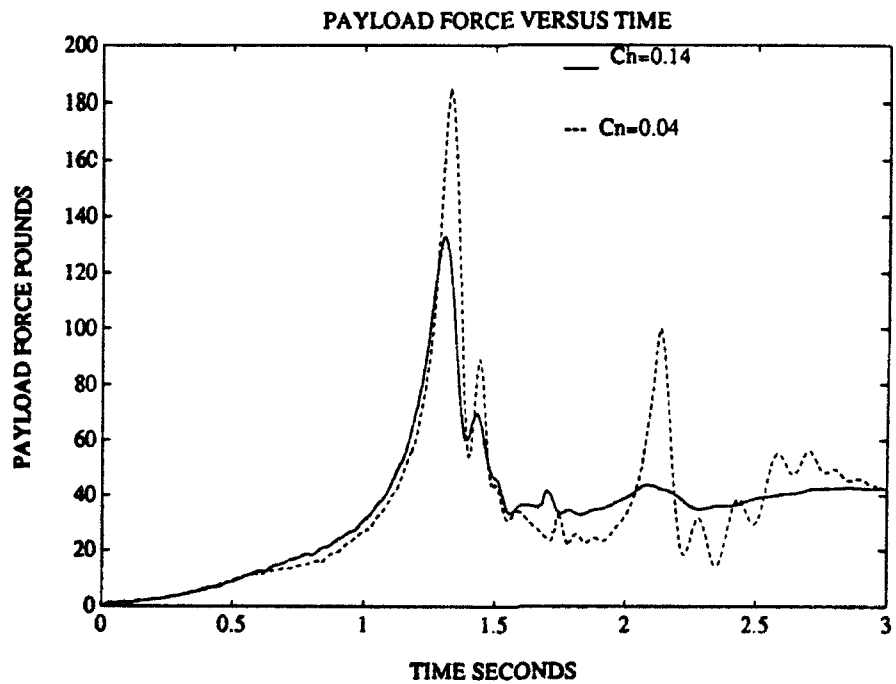


Figure A6. Payload Force Versus Time (Two Different  $C_n$ 's)

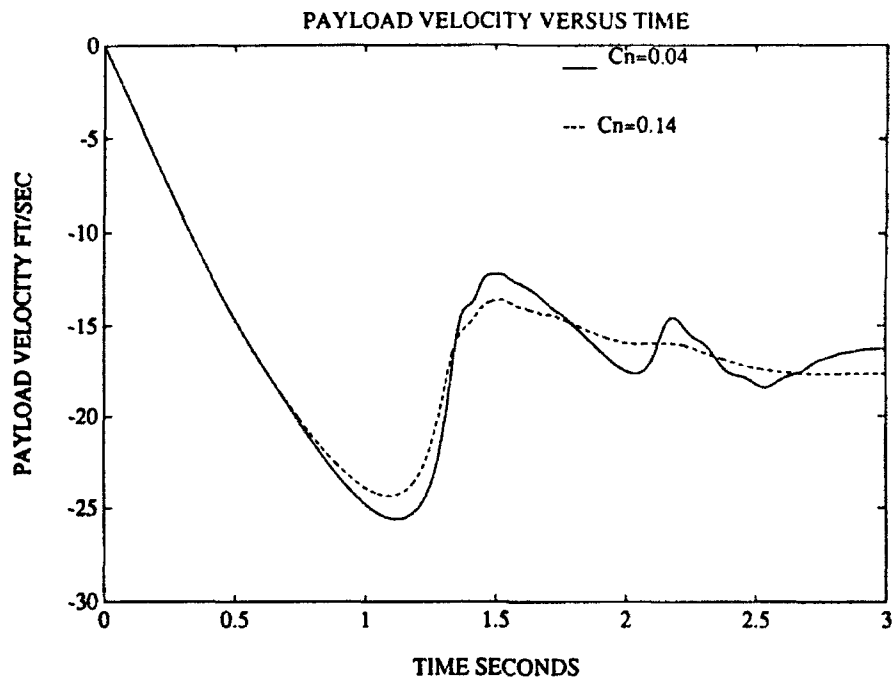


Figure A7. Payload Velocity Versus Time (Two Different  $C_n$ 's)

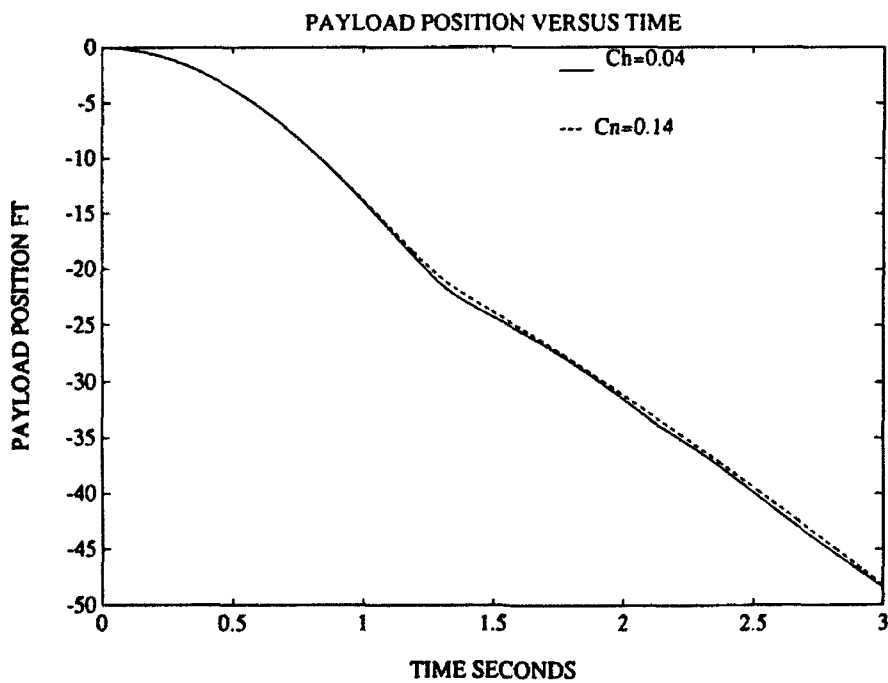


Figure A8. Payload Position Versus Time (Two Different  $C_n$ 's)

BLANK PAGE

**APPENDIX B**

**CASE 2**

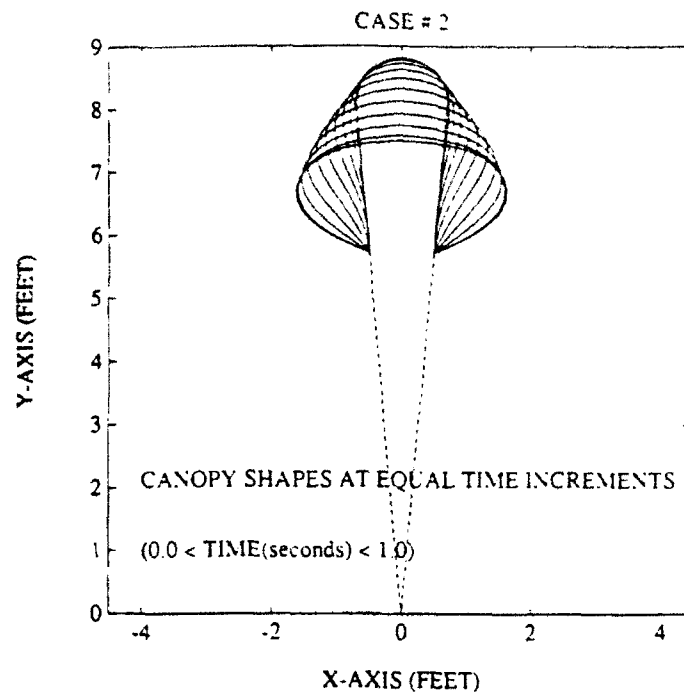


Figure B1. Canopy Shape Versus Time in Seconds ( $0.0 < t < 1.0$ )

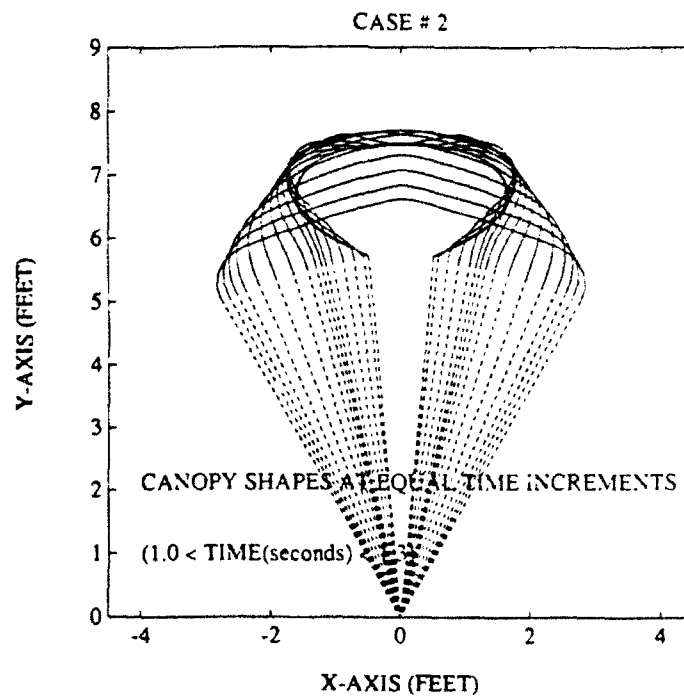


Figure B2. Canopy Shape Versus Time in Seconds ( $1.0 < t < 1.3$ )

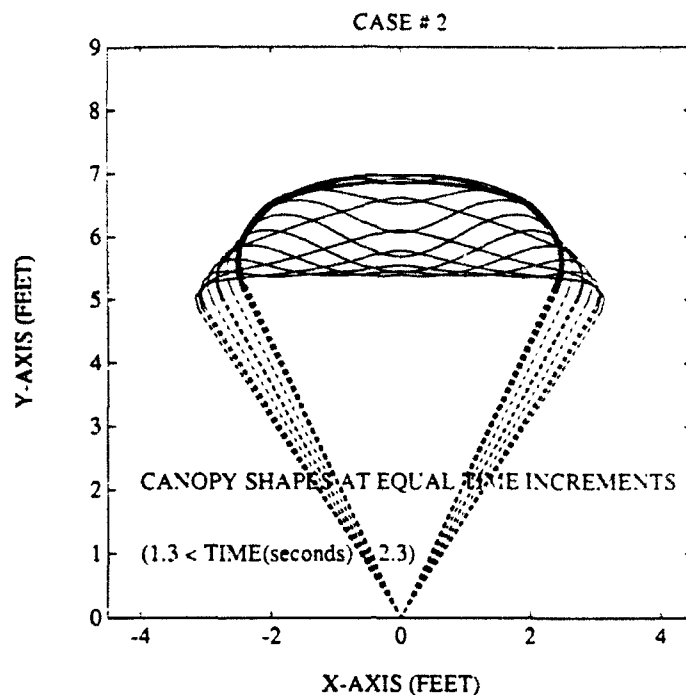


Figure B3. Canopy Shape Versus Time in Seconds ( $1.3 < t < 2.3$ )

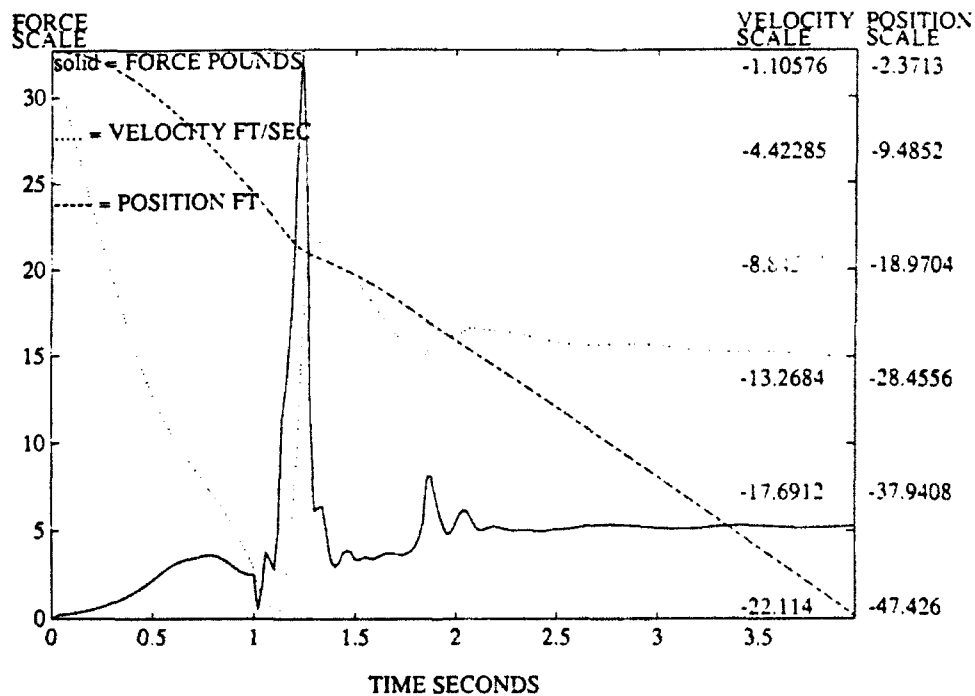


Figure B4. Payload Force, Velocity & Position Versus Time

BLANK PAGE

**APPENDIX C**

**CASE 3**



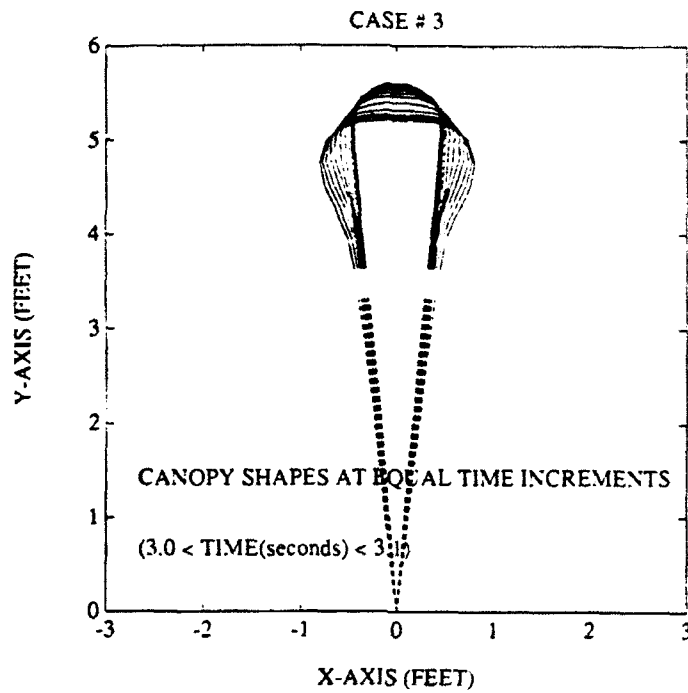


Figure C1. Canopy Shape Versus Time in Seconds ( $3.0 < t < 3.1$ )

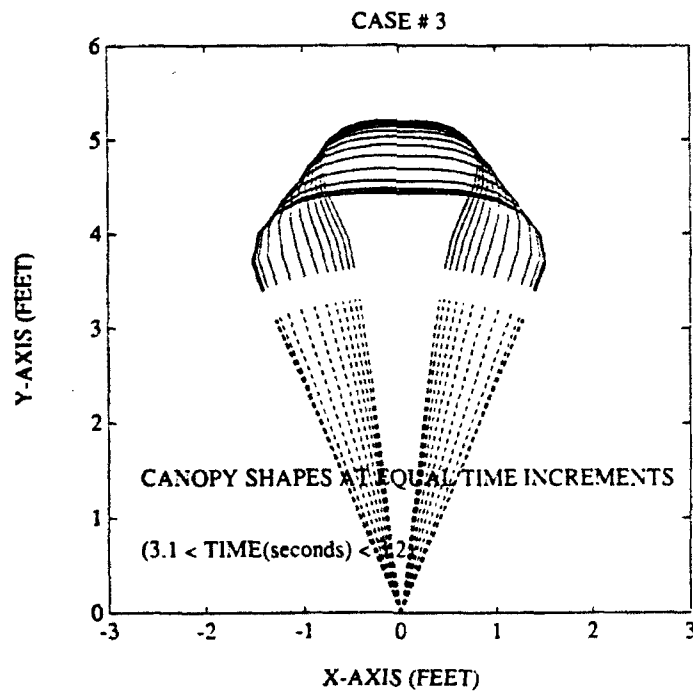


Figure C2. Canopy Shape Versus Time in Seconds ( $3.1 < t < 3.2$ )

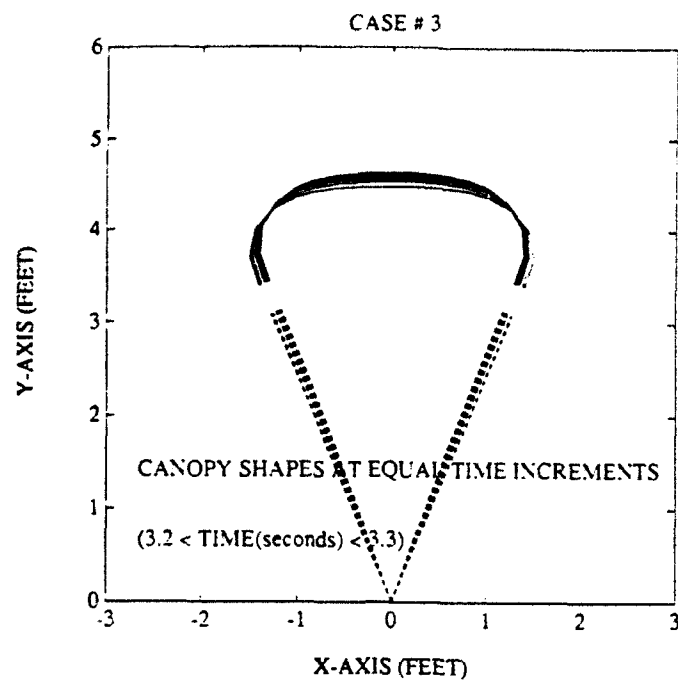


Figure C3. Canopy Shape Versus Time in Seconds ( $3.2 < t < 3.3$ )

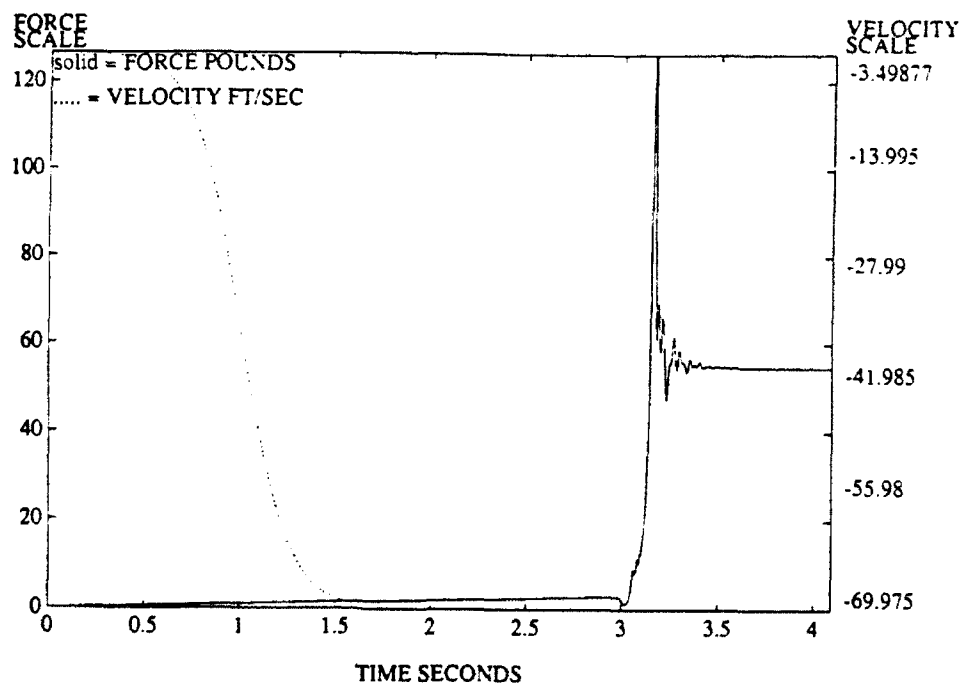


Figure C4. Payload Force and Velocity Versus Time

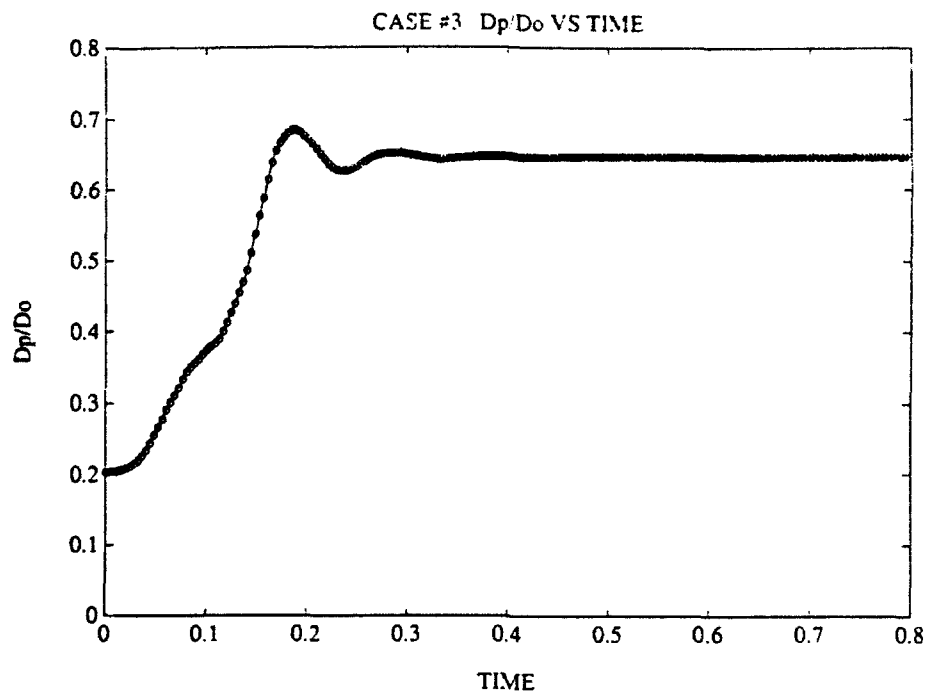


Figure C5. Projected Diameter Over Constructed Diameter vs. Time

**APPENDIX D**

**CASE 4**

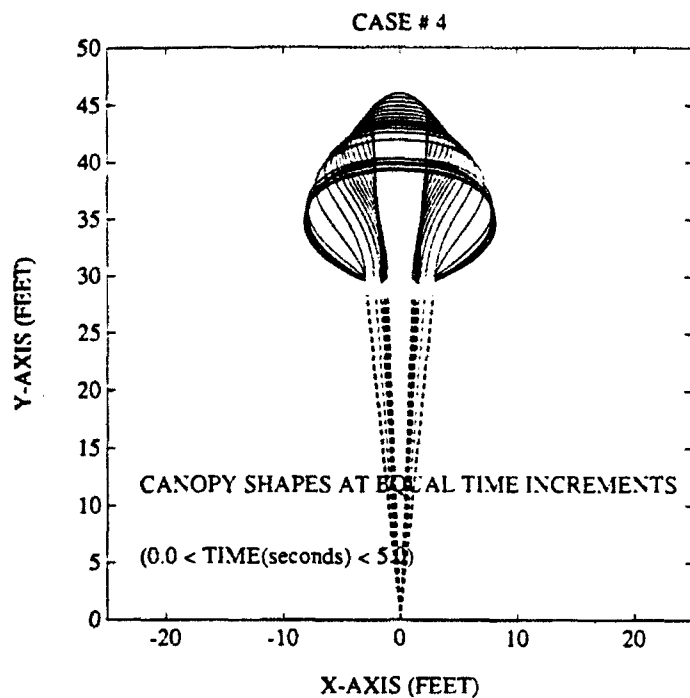


Figure D1. Canopy Shape Versus Time in Seconds ( $0.0 < t < 5.0$ )

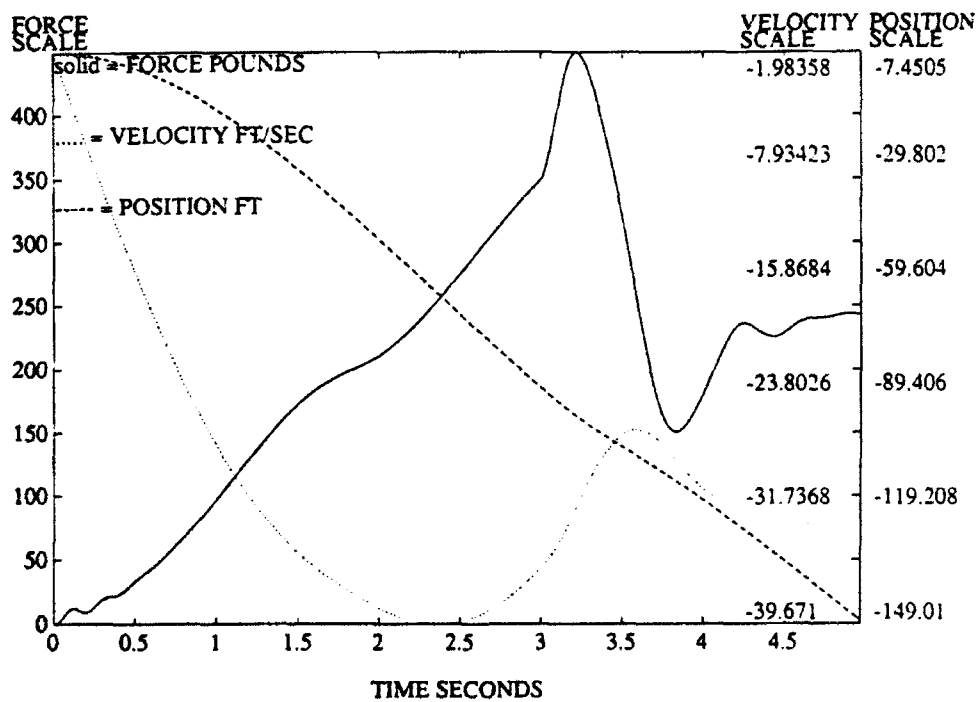


Figure D2. Payload Force, Velocity & Position Versus Time

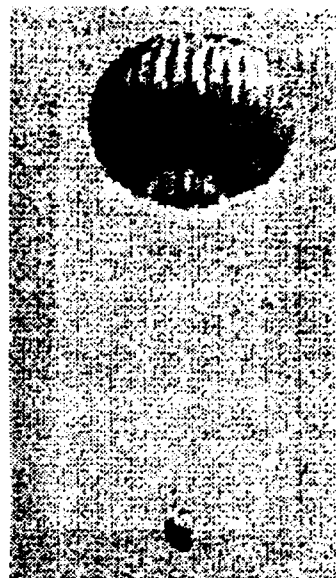
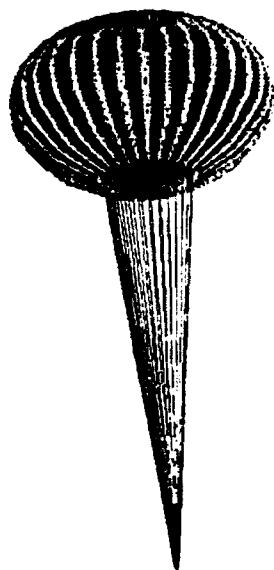


Figure D3. Images of 3-D Canopy Shapes  
(Left=Numerical,Right=Experimental)

BLANK PAGE

**APPENDIX E**

**CASE 5**



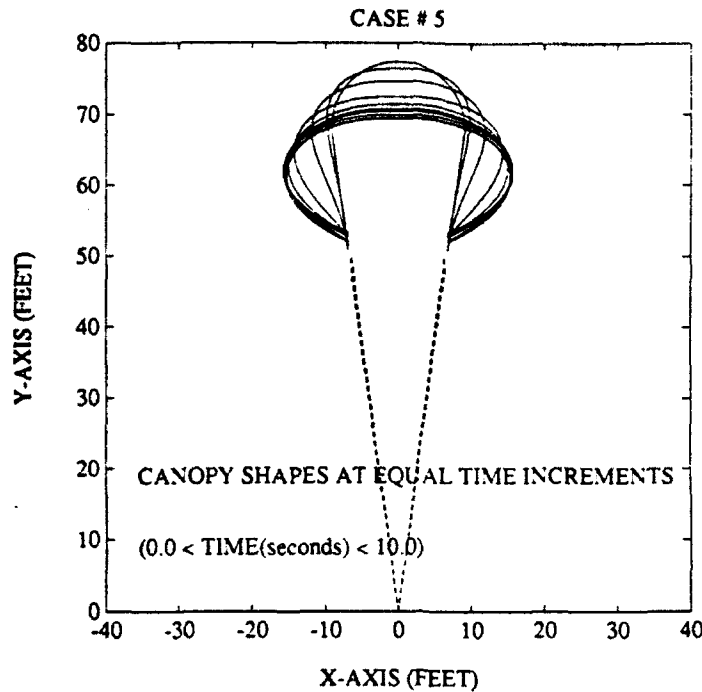


Figure E1. Canopy Shapes Versus Time in Seconds ( $0.0 < t < 10.0$ )

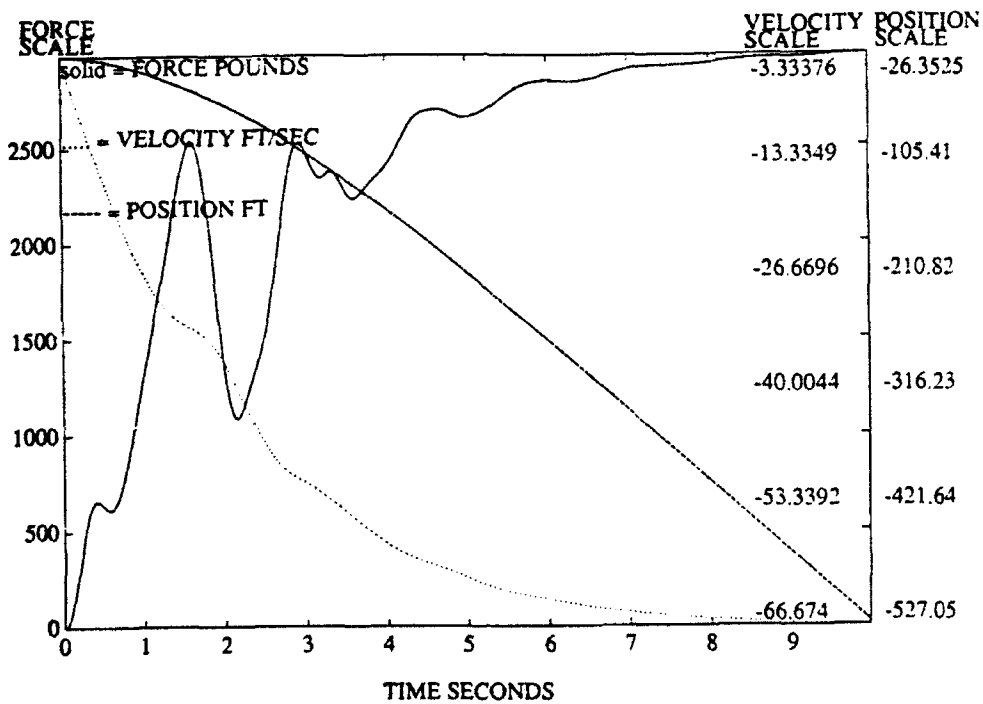


Figure E2. Payload Force, Velocity & Position Versus Time

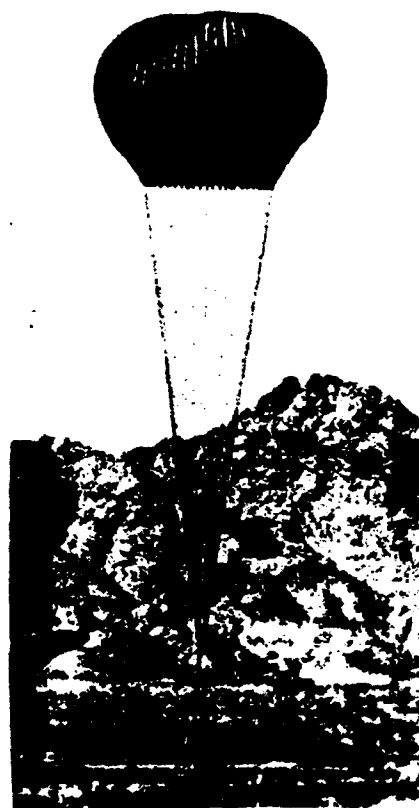
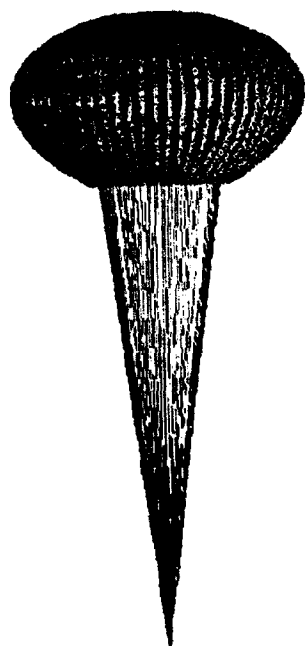


Figure E3. Images of 3-D Canopy Shapes  
(Left=Numerical,Right=Experimental)

BLANK PAGE

**APPENDIX F**

**CASE 6**

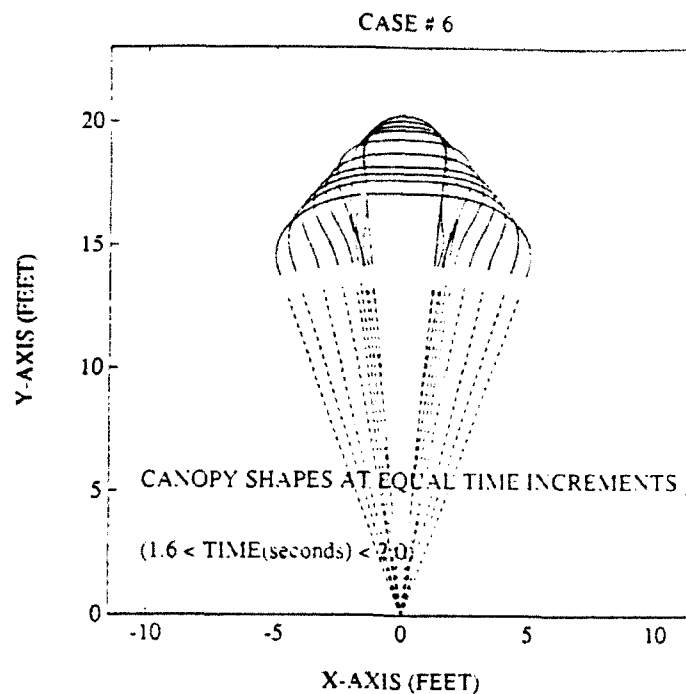


Figure F1. Canopy Shape Versus Time in Seconds ( $1.6 < t < 2.0$ )

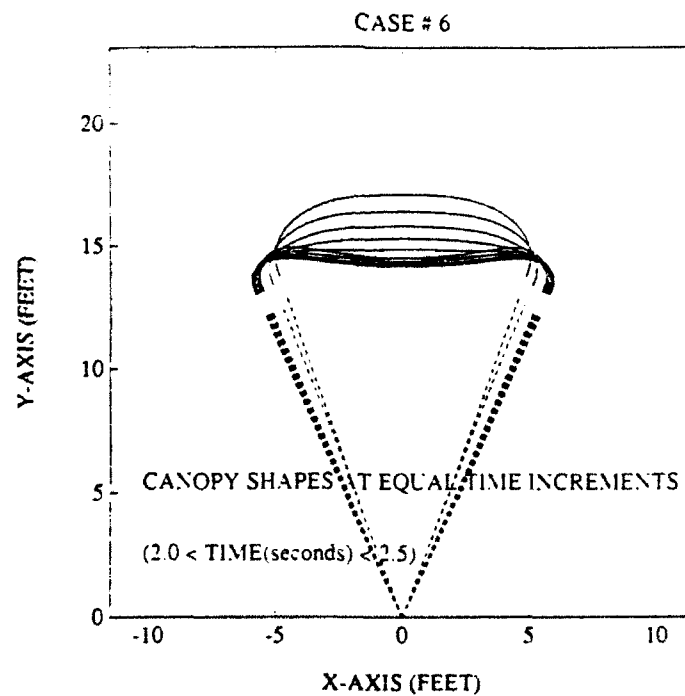


Figure F2. Canopy Shape Versus Time in Seconds ( $2.0 < t < 2.5$ )

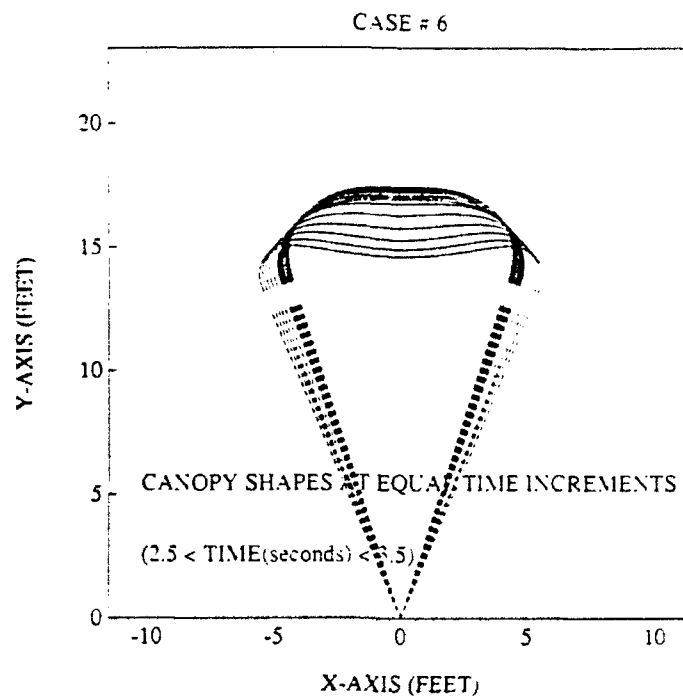


Figure F3. Canopy Shape Versus Time in Seconds ( $2.5 < t < 3.5$ )

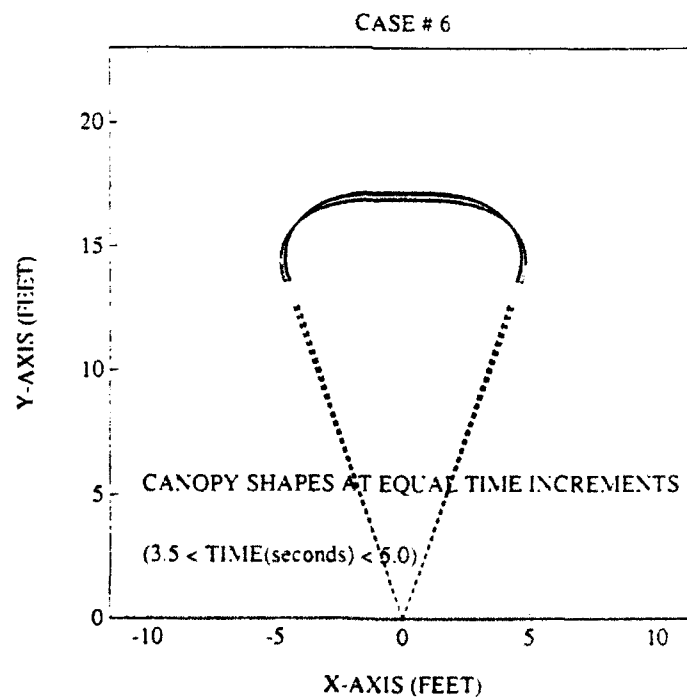


Figure F4. Canopy Shape Versus Time in Seconds ( $3.5 < t < 5.0$ )

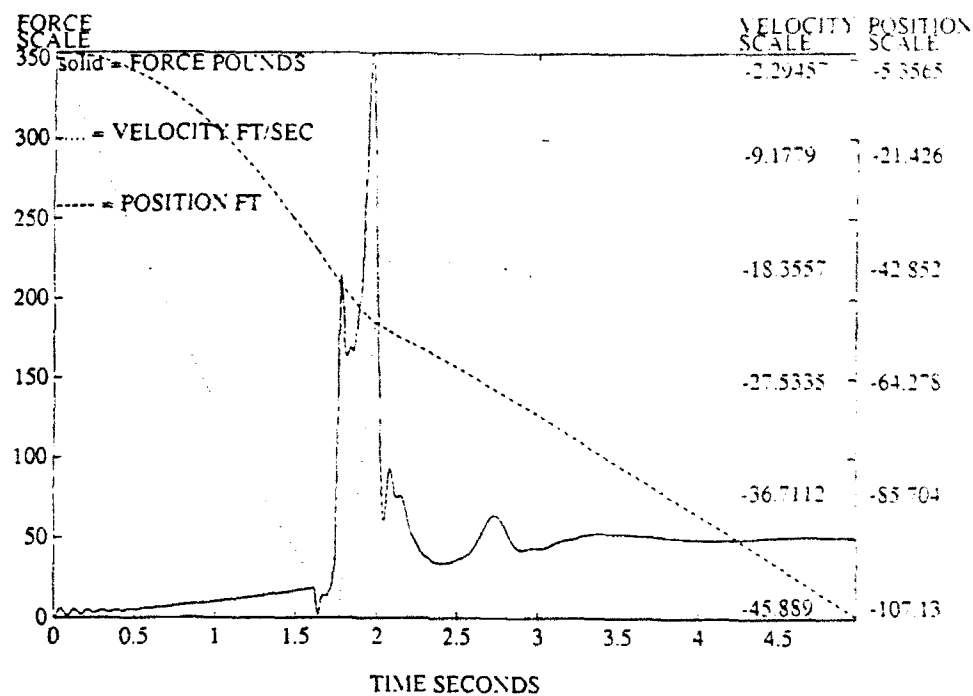


Figure F5. Payload Force, Velocity & Position Versus Time

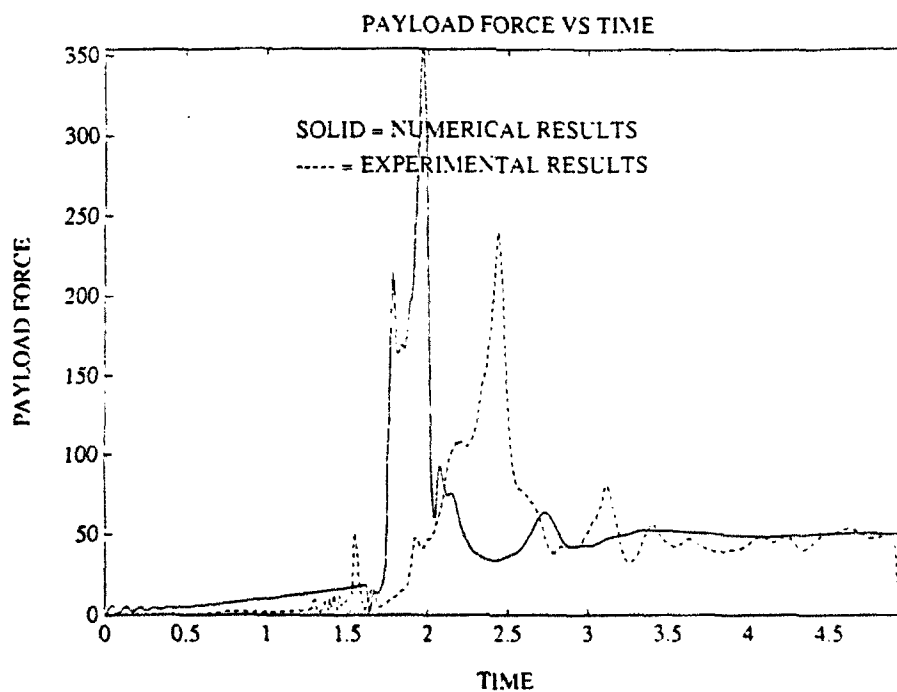


Figure F6. Payload Force Versus Time (Numerical & Experimental)

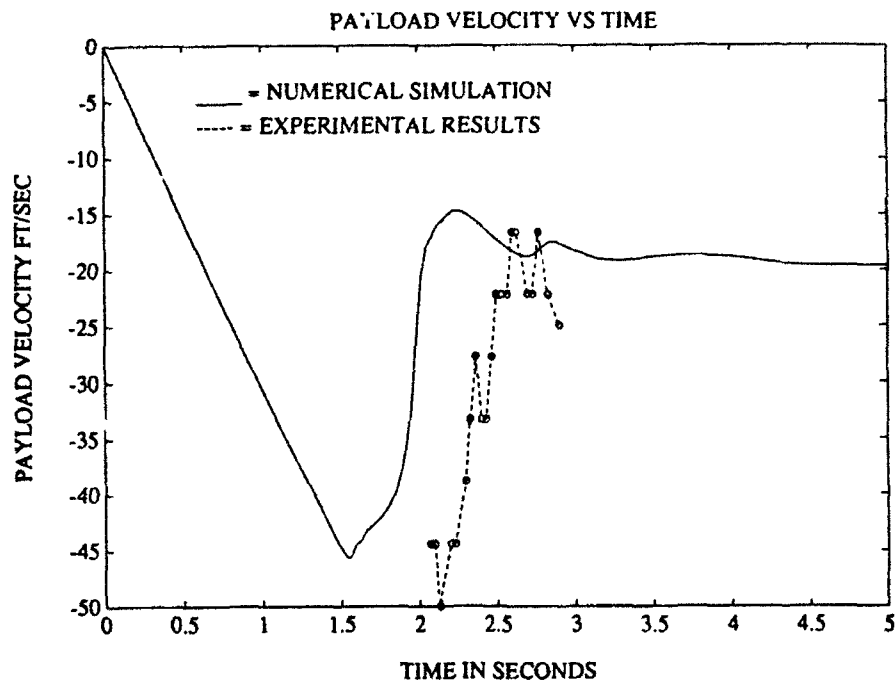


Figure F7. Payload Velocity vs. Time (Numerical & Experimental)

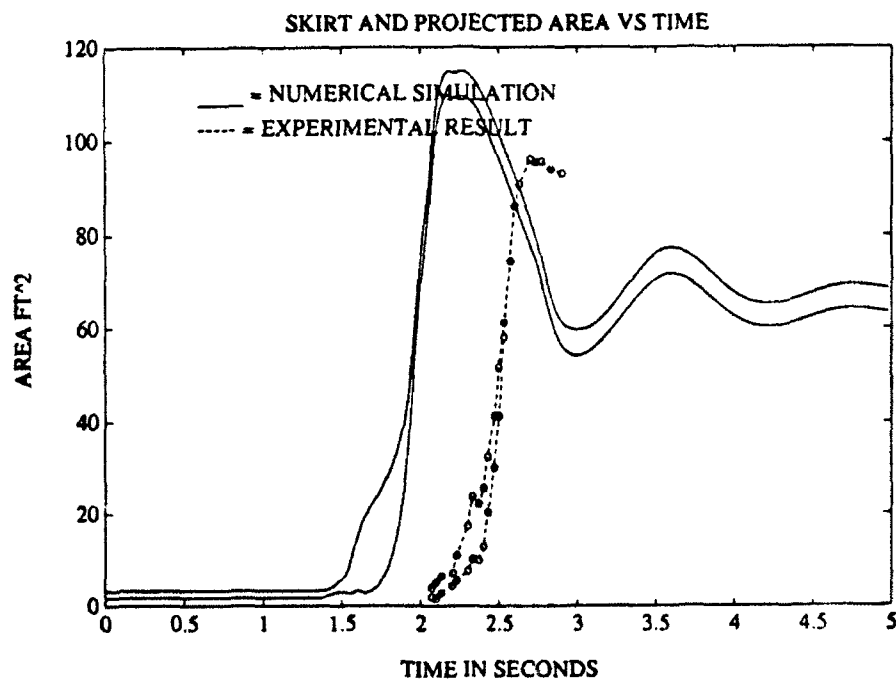


Figure F8. Canopy Area Versus Time (Numerical & Experimental)



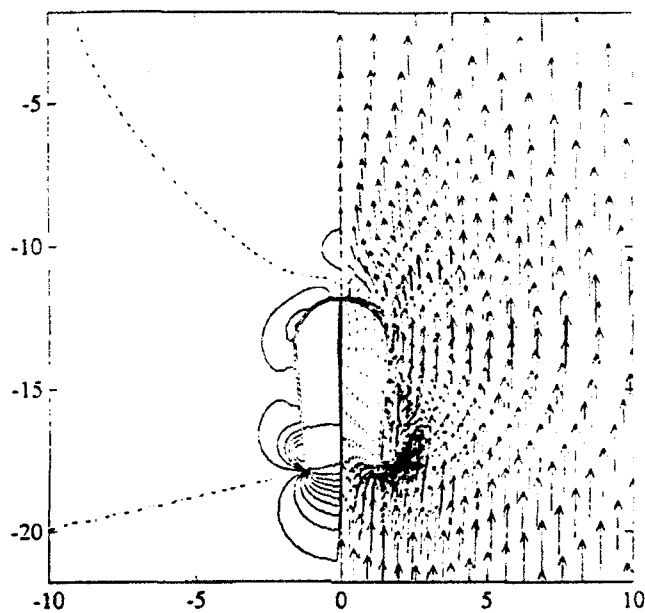


Figure F9. CFD Solution ( $t=1.58$  seconds)

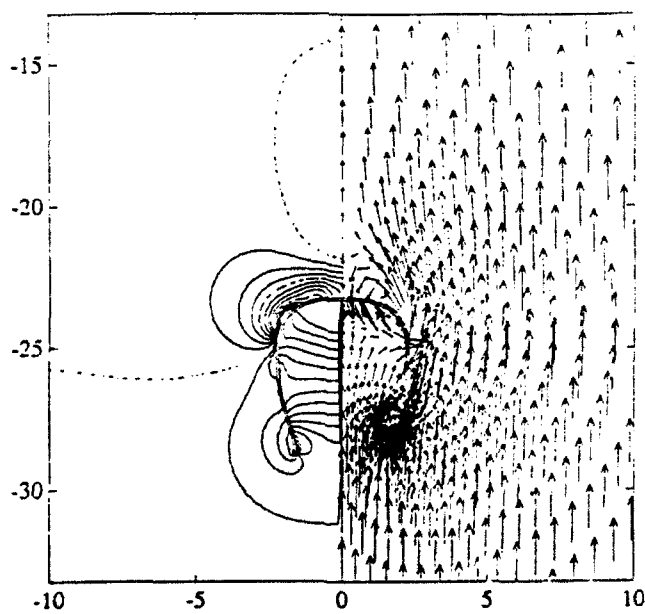


Figure F10. CFD Solution ( $t=1.84$  seconds)

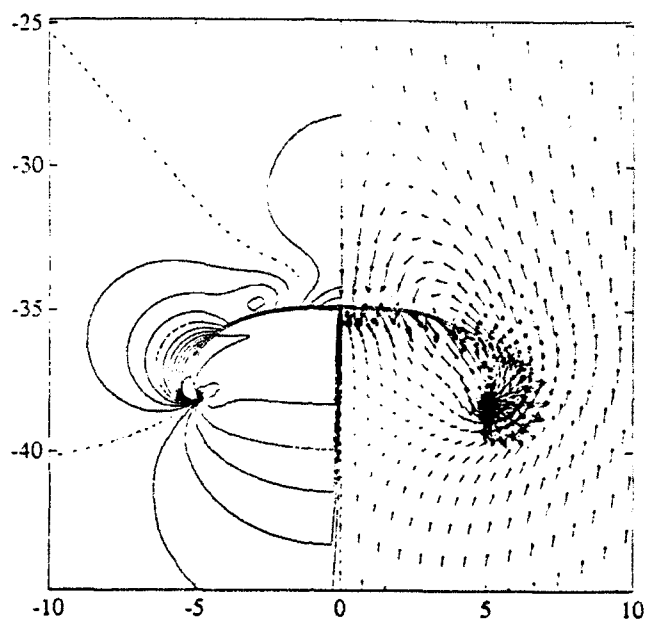


Figure F11. CFD Solution ( $t=2.11$  seconds)

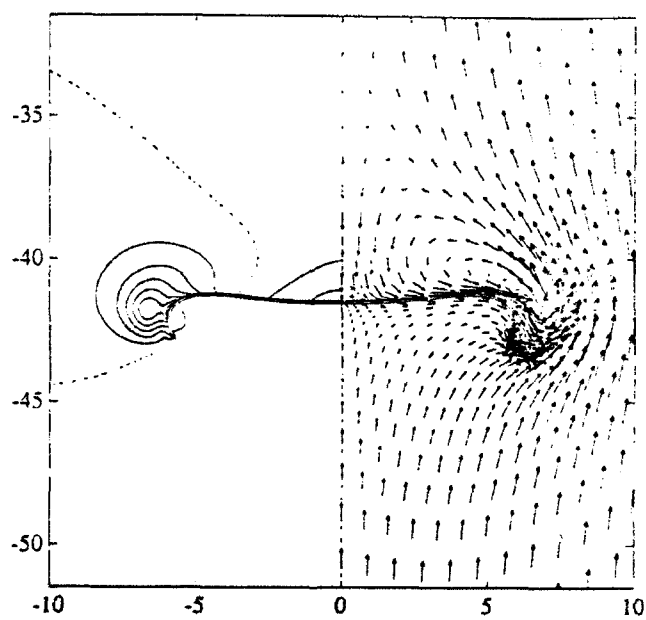


Figure F12. CFD Solution ( $t=2.37$  seconds)

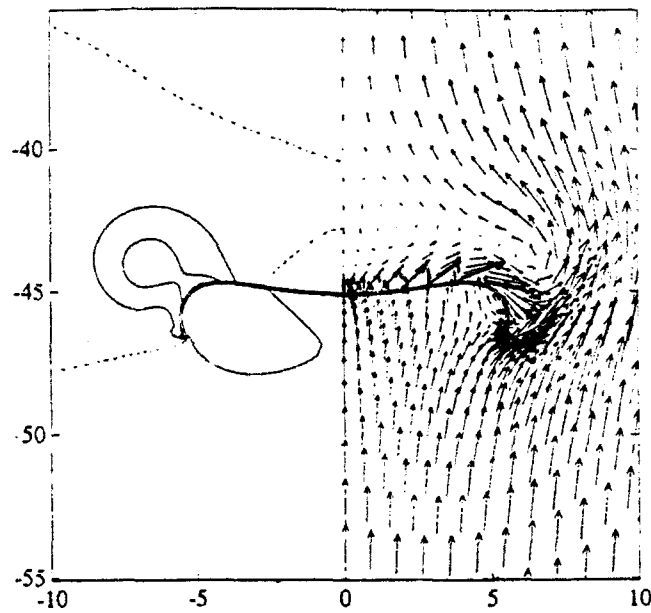


Figure F13. CFD Solution ( $t=2.63$  seconds)

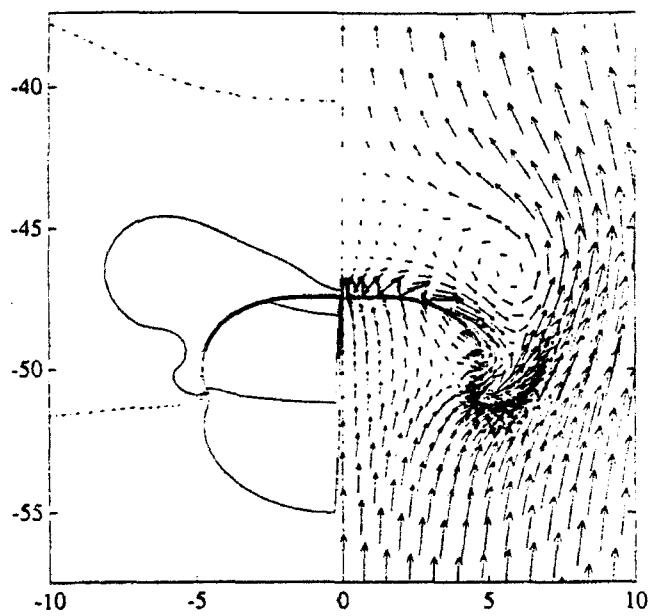


Figure F14. CFD Solution ( $t=2.89$  seconds)

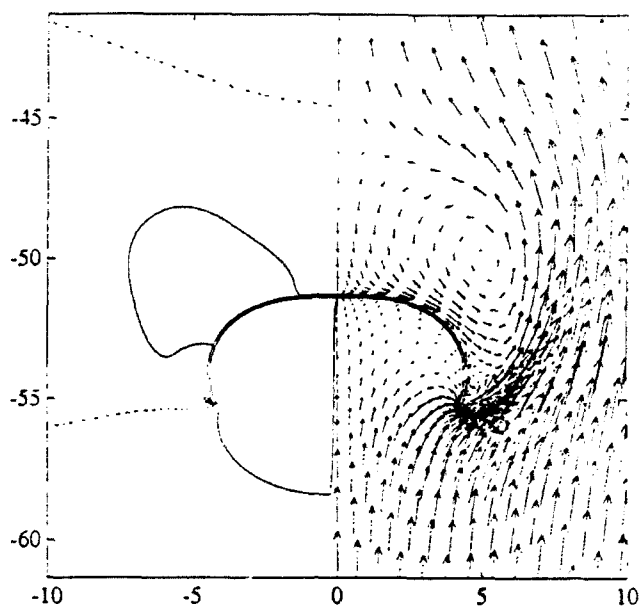


Figure F15. CFD Solution ( $t = 3.16$  seconds)

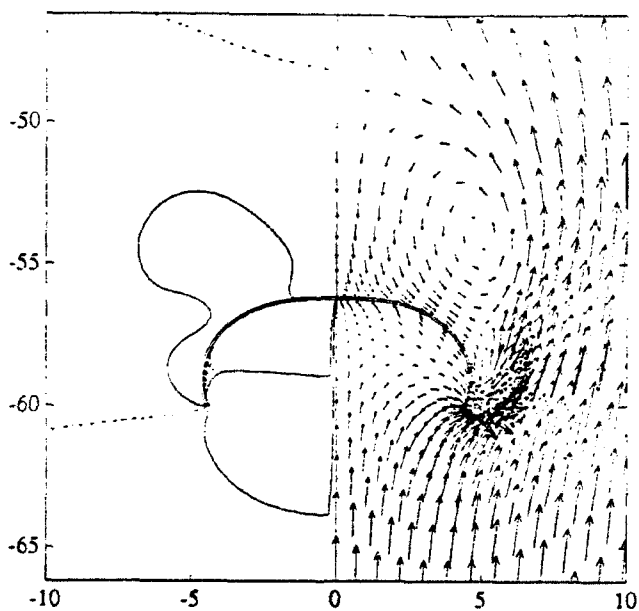


Figure F16. CFD Solution ( $t = 3.42$  seconds)

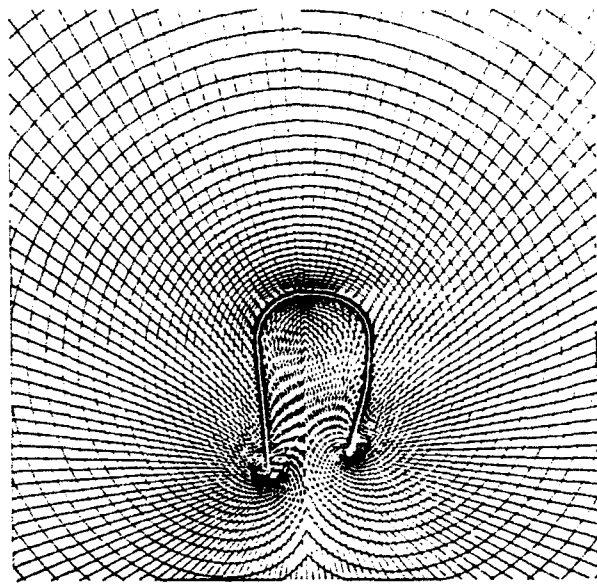


Figure F17. CFD Grid ( $t=1.58$  &  $1.84$  seconds)

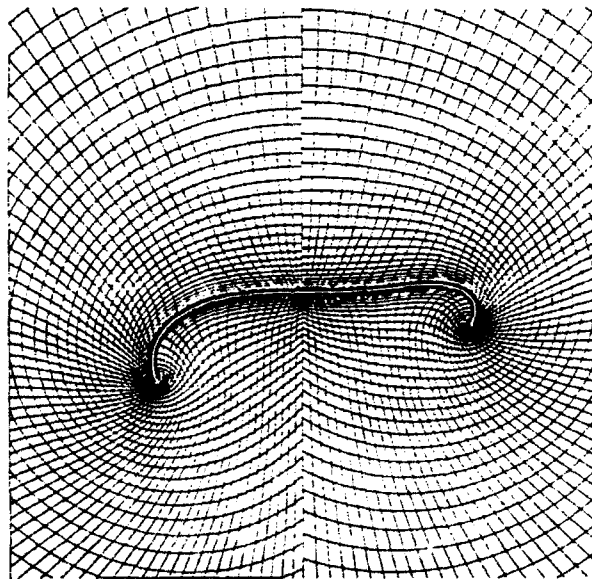


Figure F18. CFD Grid ( $t= 2.11$  &  $2.37$  seconds)

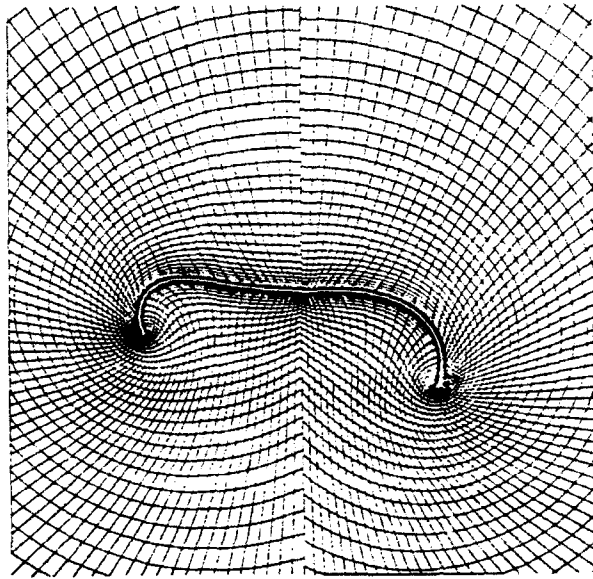


Figure F19. CFD Grid ( $t=2.63$  &  $2.89$  seconds)

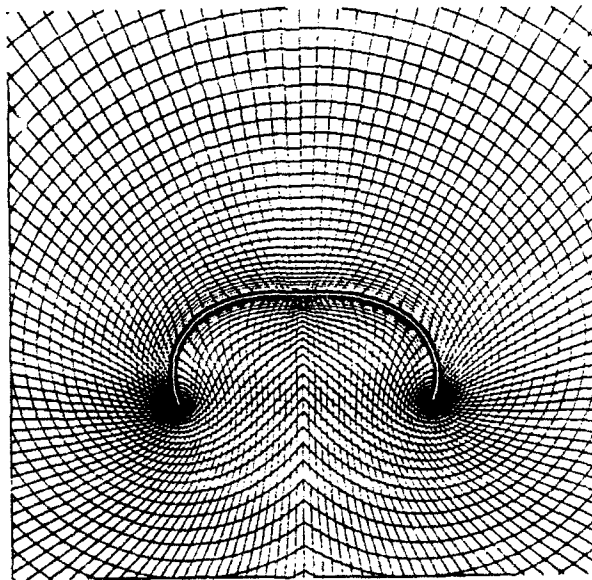


Figure F20. CFD Grid ( $t=3.16$  &  $3.42$  seconds)